|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № \_\_3\_\_**

|  |  |
| --- | --- |
| **Тема ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ОСНОВНЫХ АЛГОРИТМОВ ПОСТРОЕНИЯ ОТРЕЗКОВ И ИССЛЕДОВАНИЕ ИХ ВРЕМЕННЫХ И ВИЗУАЛЬНЫХ ХАРАКТЕРИСТИК**  **Студент \_Брянская Е.В.\_\_\_\_\_\_\_\_**  **Группа \_ИУ7-42Б\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_Куров А.В.** |  |

Москва.

2020 г.

**Цель работы:** реализация алгоритмов построения отрезков по методу цифрового дифференциального анализатора (ЦДА), алгоритмов Брезенхема (действительного, целочисленного и с устранением ступенчатости), алгоритма Ву, а также встроенной функции построения отрезка и исследование их характеристик, и сравнение полученных результатов.

Задание: в ходе выполнения практической части этой лабораторной работы необходимо выполнить следующие пункты:

1. Реализовать алгоритмы ЦДА, Брезенхема (действительный, целочисленный, с устранением ступенчатости), Ву
2. Сравнить визуально отрезки, построенные в соответствии с каждым алгоритмом, а также с отрезком, построенным процедурой языка высокого уровня. Проверить попадание отрезка в заданную конечную точку.
3. Определить время, затрачиваемое на построение отрезка по каждому из алгоритмов.
4. Для заданного алгоритма получить зависимость длины максимальной ступеньки от угла наклона отрезка и отобразить ее в виде графика или гистограммы.

Программа, реализующие алгоритмы, должна обеспечивать построение произвольного отрезка, а также предусматривать высвечивание точки в случае вырожденного отрезка. Необходимо предоставить возможность пользователю выбирать цвет строящегося отрезка (предусмотреть рисование фоновым цветом).

Необходимо сделать проверку на попадание отрезка в требуемую конечную точку.

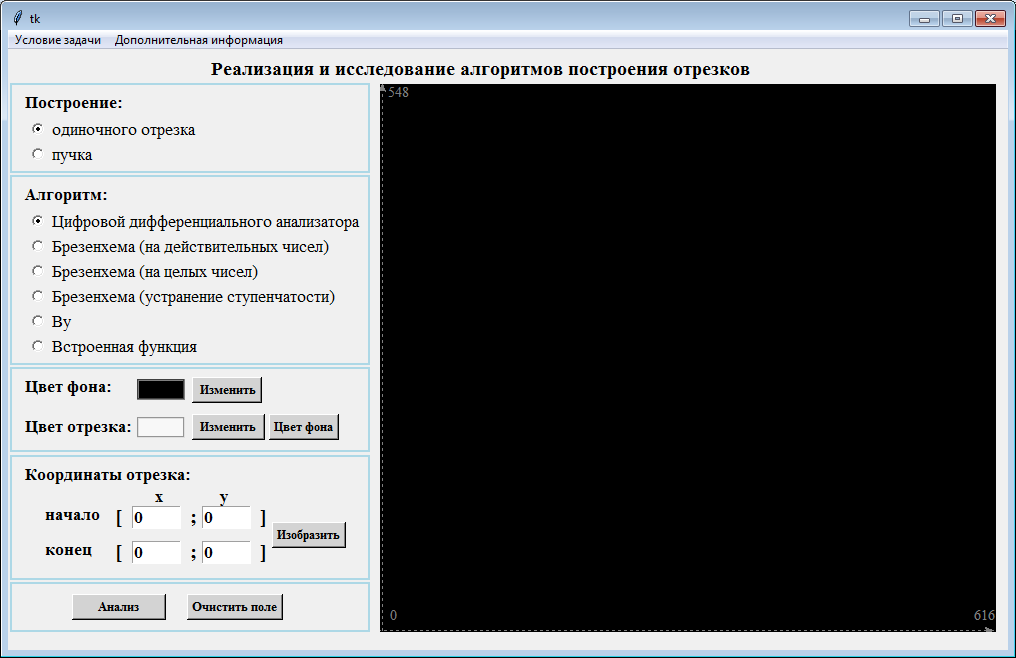
При анализе визуальных характеристик нужно отрисовывать пучок лучей, путём задания пользователем его центра, длины лучей и шаг изменения узла.

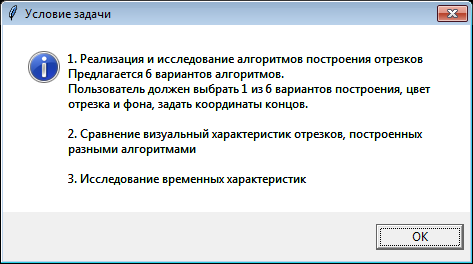
Общие требования к алгоритмам:

1. отрезки должны выглядеть прямыми, начинаться и заканчиваться в заданных точках
2. яркость вдоль отрезка должна быть постоянной и не зависеть от длины и наклона
3. алгоритмы должны работать быстро

**Интерфейс**

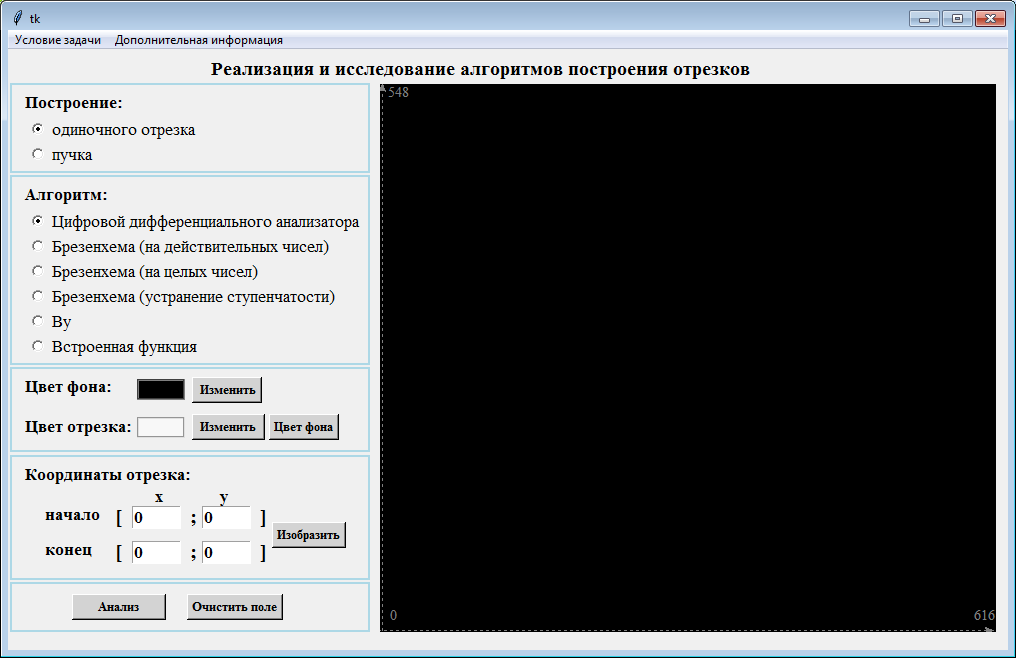
Пользователю предоставляется следующий интерфейс для работы:





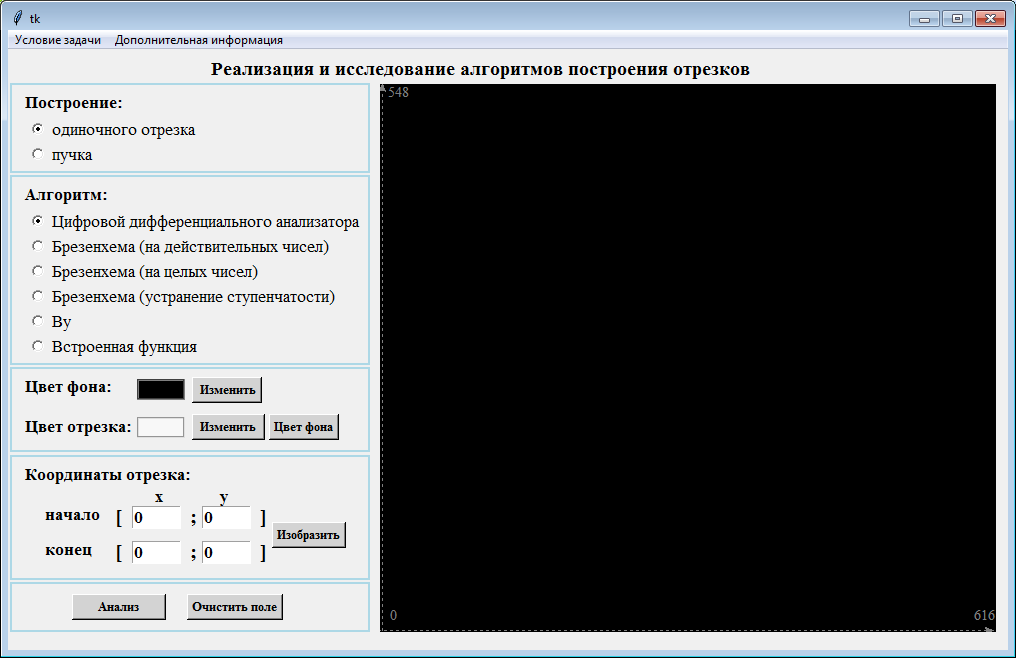
Он предоставляет возможность просмотреть условие задачи, а также получить дополнительную информацию о программе:

Перед началом работы пользователь выбирает режим:

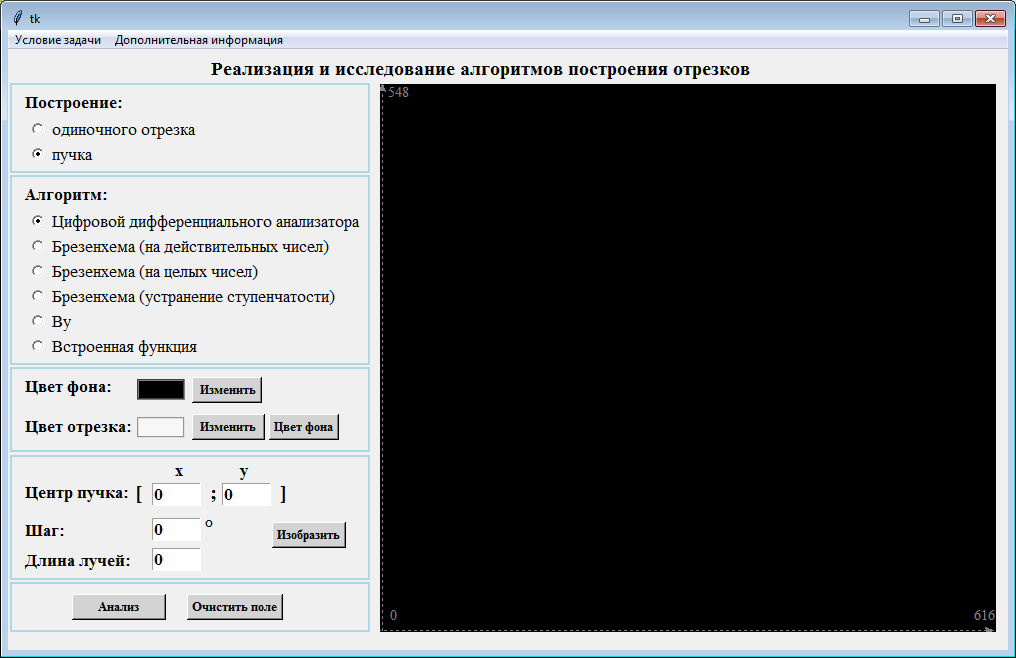


Причём в соответствии с режимом появляется необходимое меню.

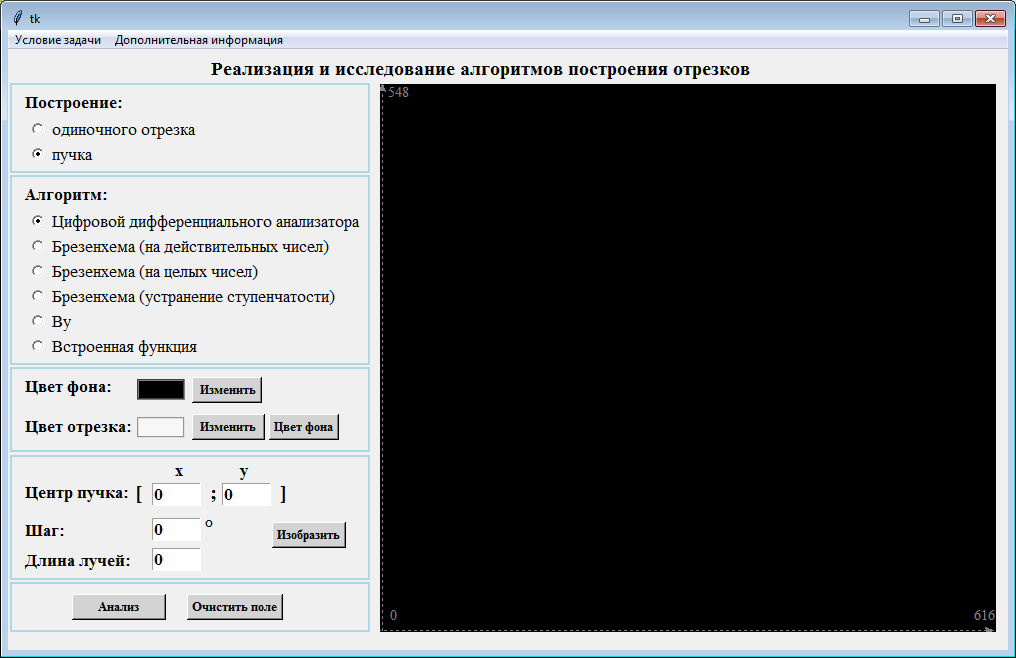
Так для режима «одиночного отрезка» меню выглядит следующим образом:

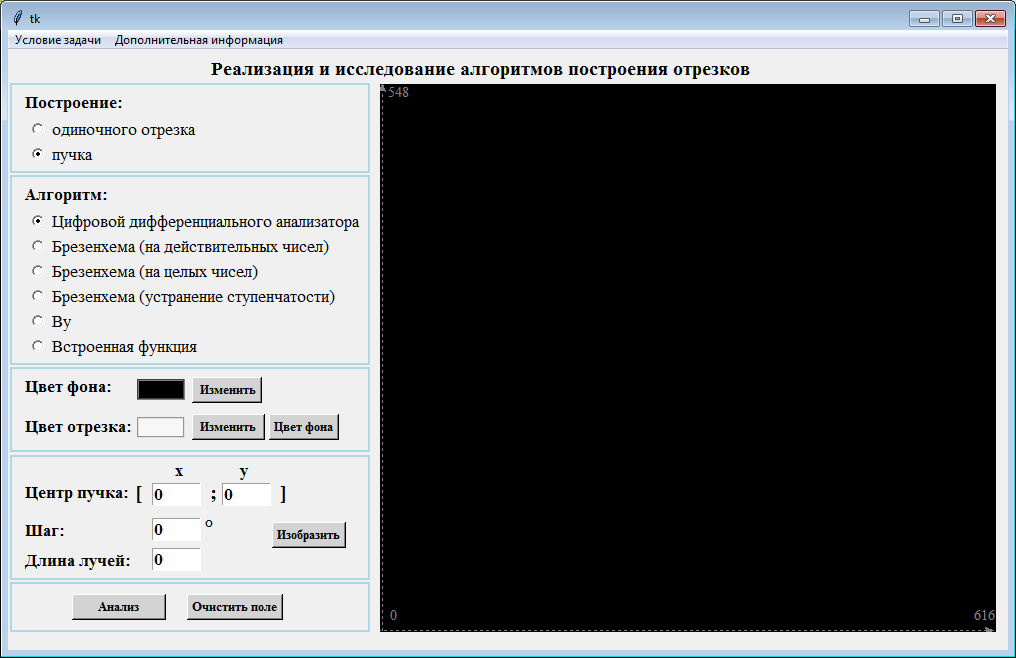
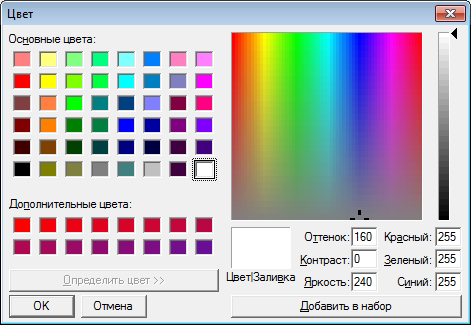


А для режима «пучка» это меню сменяется другим:

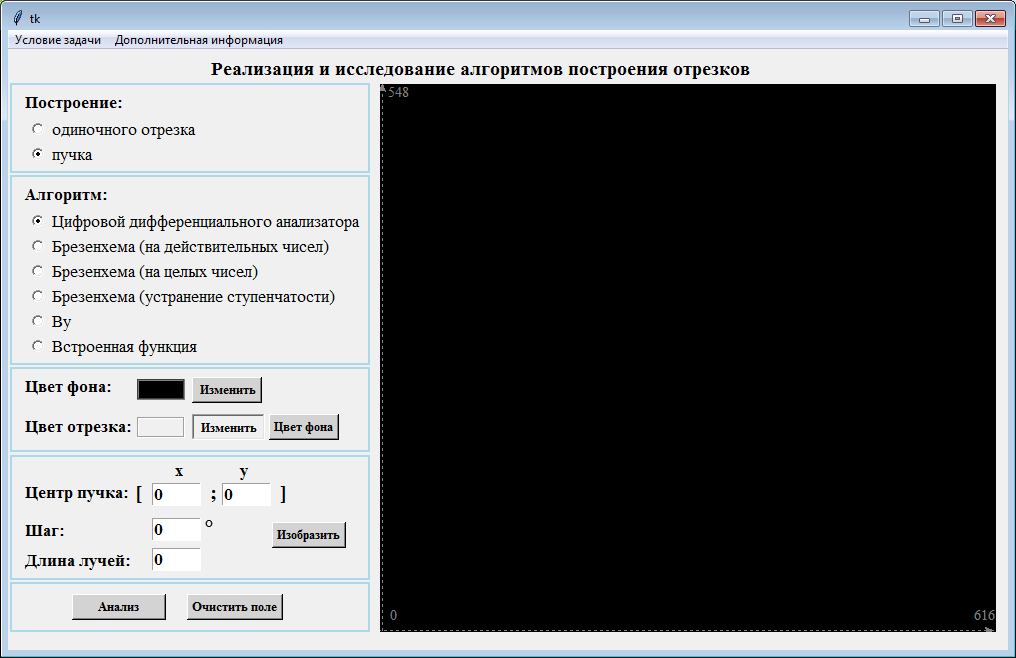


Интерфейс предлагает пользователю выбрать алгоритм построения (вне зависимости от выбранного режима работы):



А также можно выбрать цвет отрезка и фона при нажатии на соответствующую кнопку (для отрезка предусмотрена дополнительная кнопка рисования цветом фона). По умолчанию цвет фона – чёрный, отрезка – белый.

Ещё предусмотрены такие кнопки, как «Очистить поле» и «Анализ».



**Рассматриваемые алгоритмы**

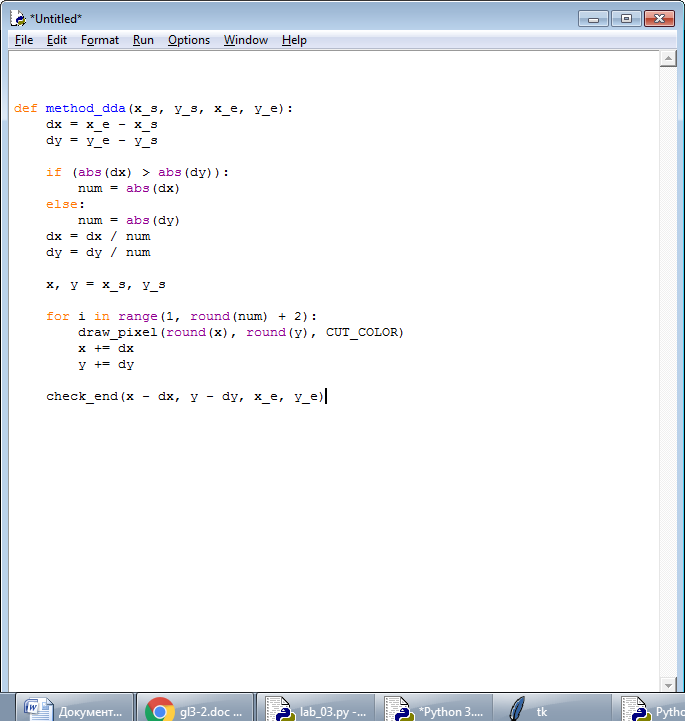
1. Алгоритм цифрового дифференциального анализатора (ЦДА)

Этот алгоритм использует достаточно общий принцип, известный в математике: использование дифференциального уравнения или системы таких уравнений.

Решением которого является:

(при ∆х = 1)

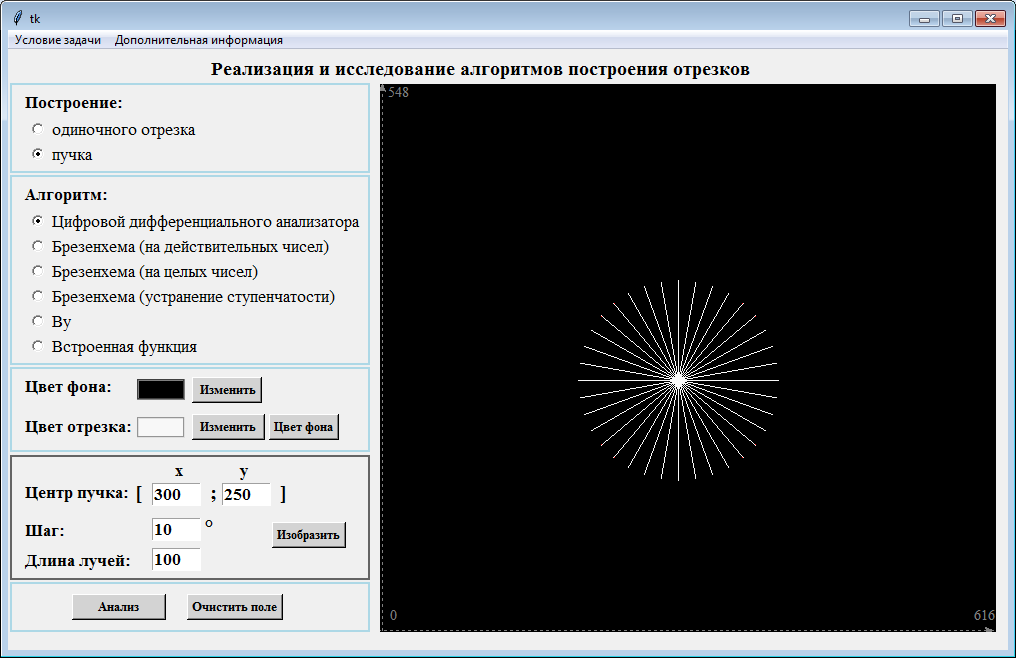
*Программная реализация:*



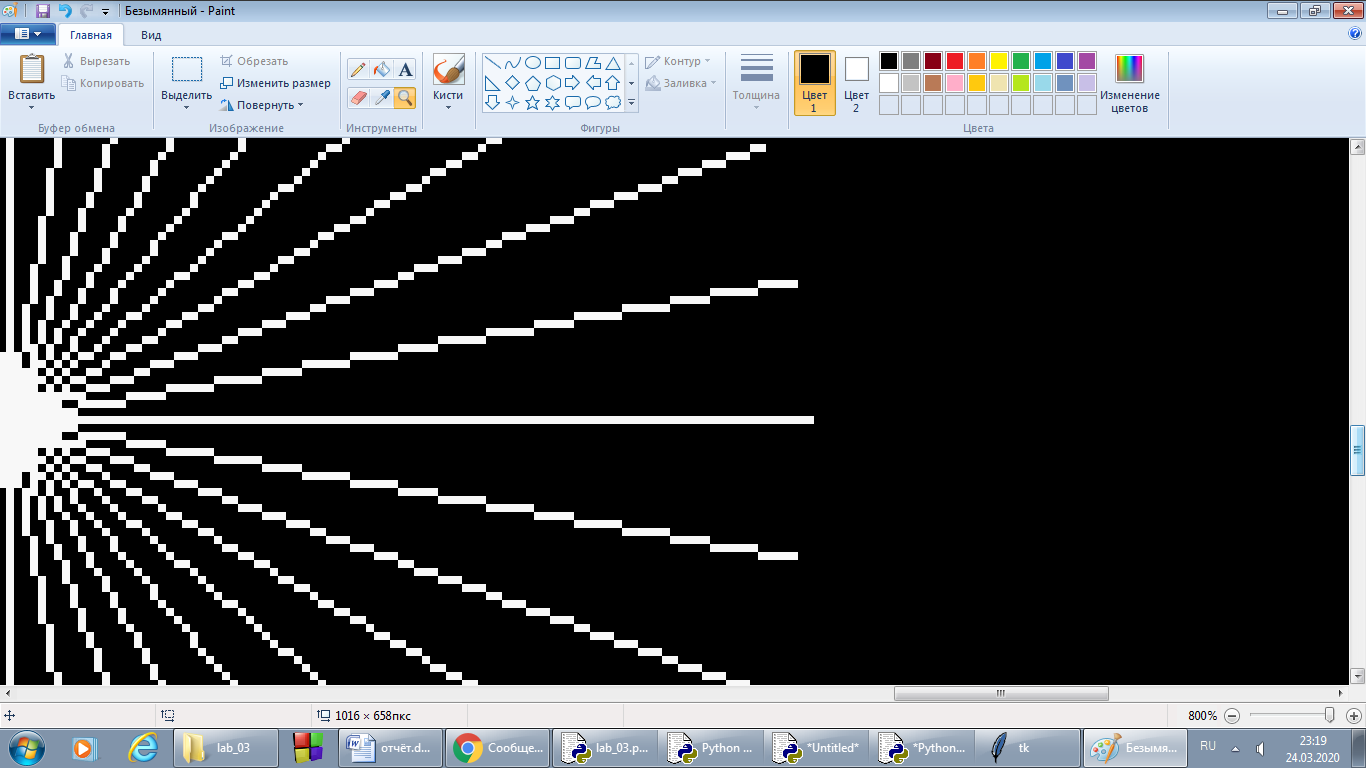
*Минусы данного алгоритма:*

1. медленный
2. точность в концевых точках уменьшается (большая вероятность того, что отрезок не попадёт в нужную точку)

*Пример работы программы:*



Если воспользоваться увеличением, то можно увидеть следующую картину:



1. Алгоритм Брезенхема на действительных числах

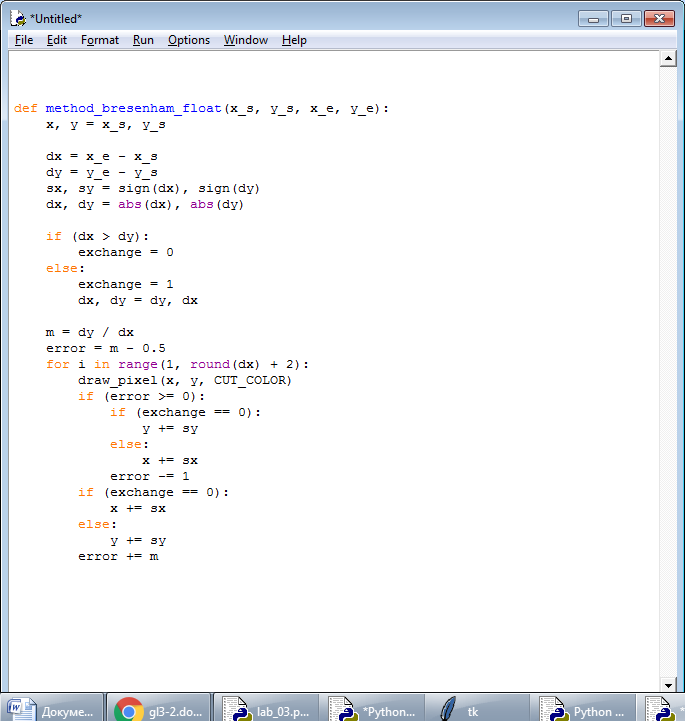
Работа алгоритма Брезенхема основывается на использовании понятия «ошибка». Ошибкой называется расстояние между действительным положением отрезка и ближайшим пикселом сетки растра, который

аппроксимирует отрезок на очередном шаге.

На каждом шаге одна из координат - либо x, либо y (в зависимости от угла наклона отрезка) изменяется на единицу, изменение другой координаты определяется в процессе работы и зависит от значения ошибки.

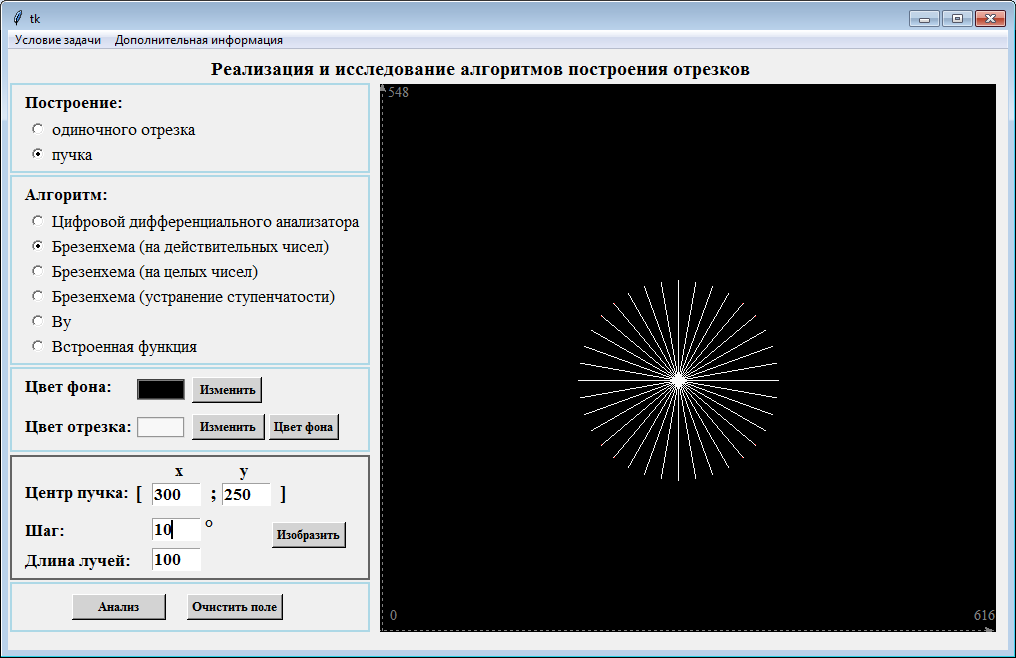
Таким образом, выбирается пиксел, ближе расположенный к идеальному отрезку.

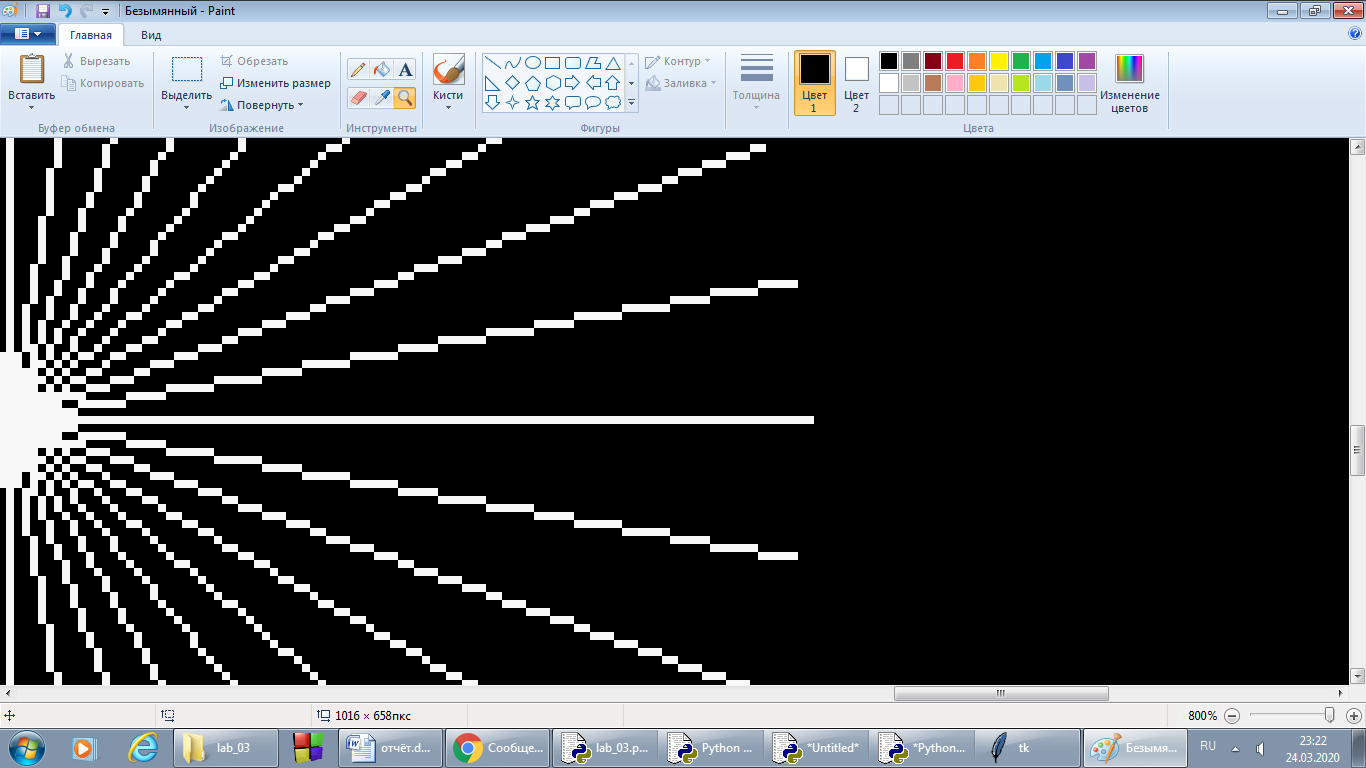
*Программная реализация:*



*Минусы данного алгоритма:*

1. использование вычислений с плавающей точкой
2. использование операции деления при определении тангенса угла наклона отрезка (сократить время работы алгоритма можно при использовании целочисленной арифметики и отказа от операции деления)

*Пример работы программы:*

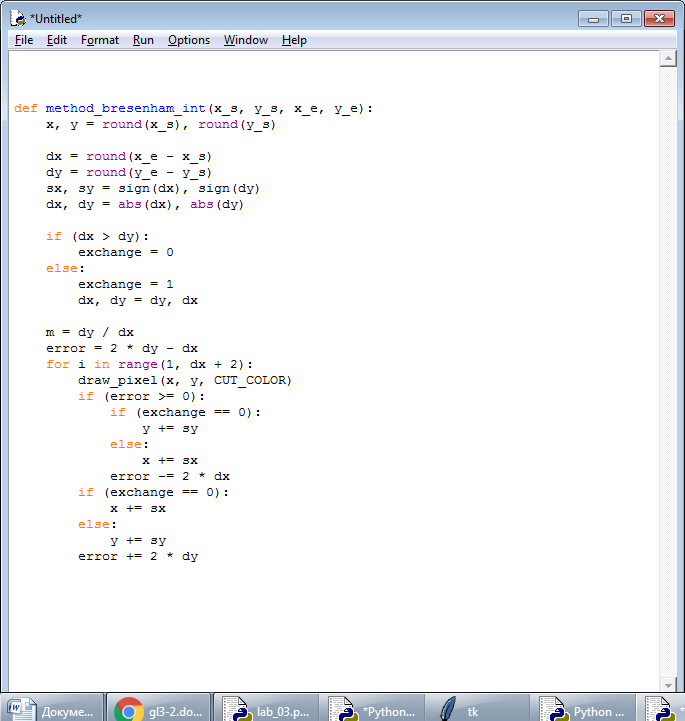
Если увеличить изображение, то оно будет схожим с предыдущим алгоритмом:

1. Алгоритм Брезенхема на целых числах

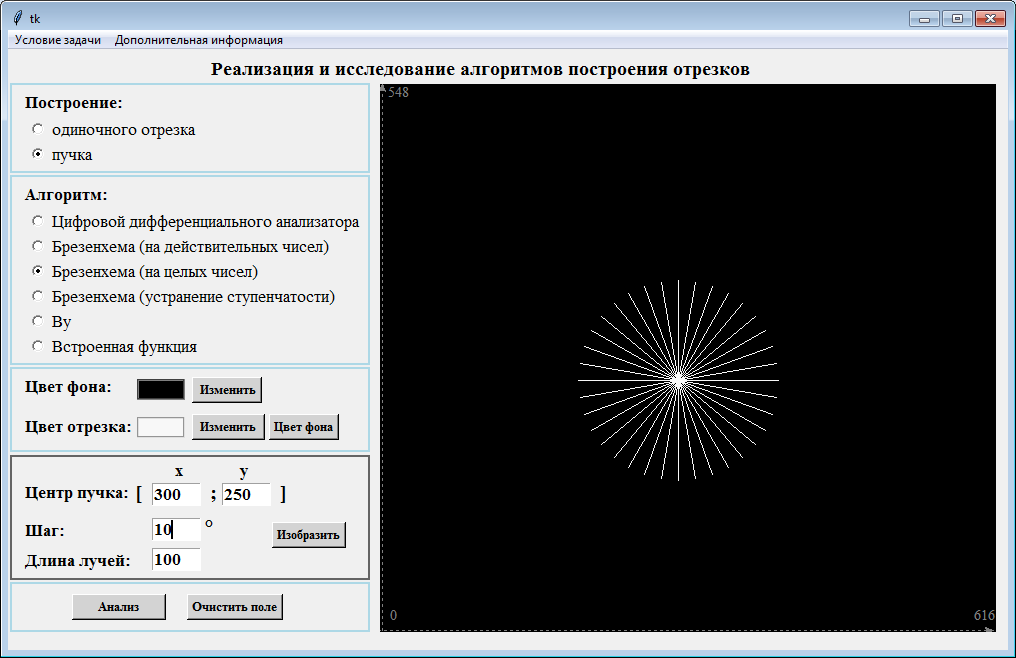
Этот алгоритм призван сократить время работы путём использования целочисленной арифметики и отказа от операции деления.

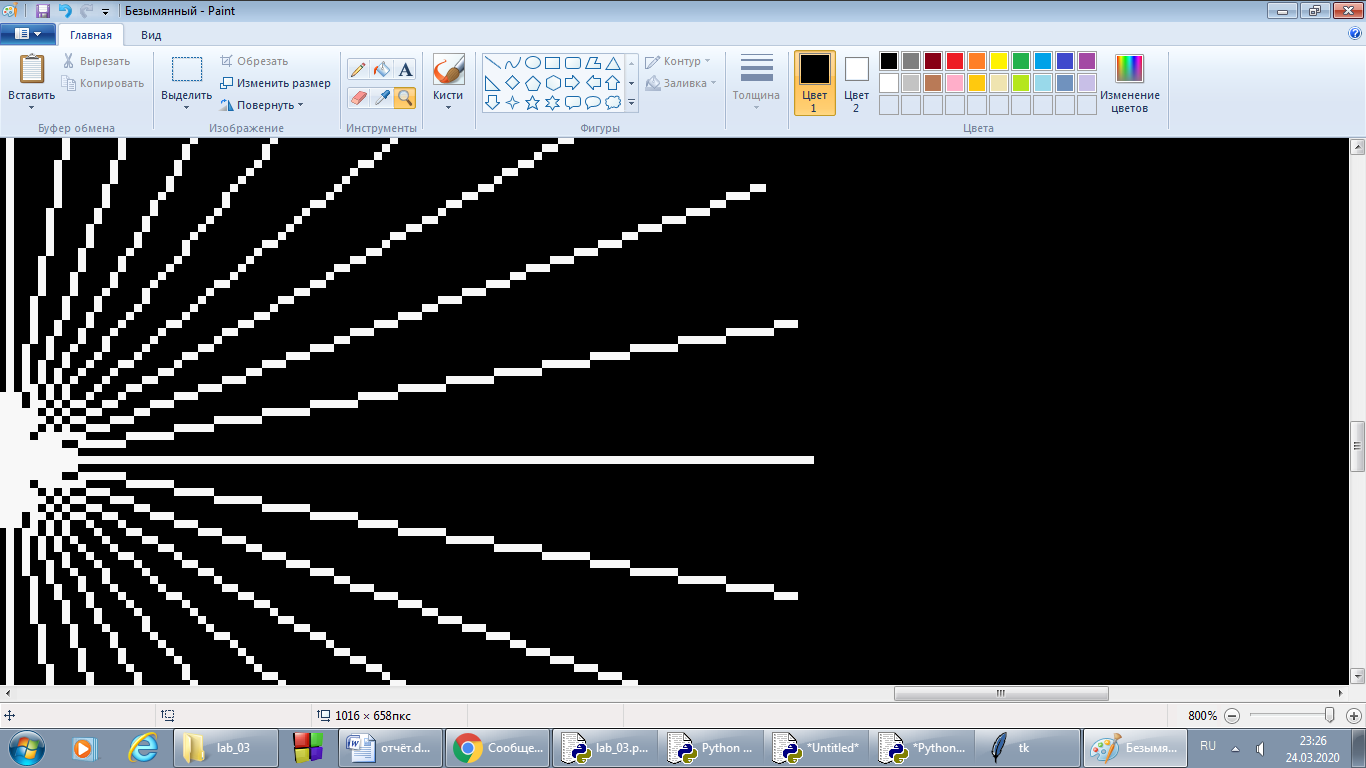
Преобразовать предыдущий алгоритм к целочисленному варианту достаточно легко. Чтобы этого достичь нужно умножить все вычисления, связанные с ошибкой на **2dx**. То есть изначально ошибка задаётся как **error = 2dy – 1**,а с каждым шагом увеличивается на **2dy**.

*Программная реализация:*

**

*Пример работы программы:*

**

Увеличение:

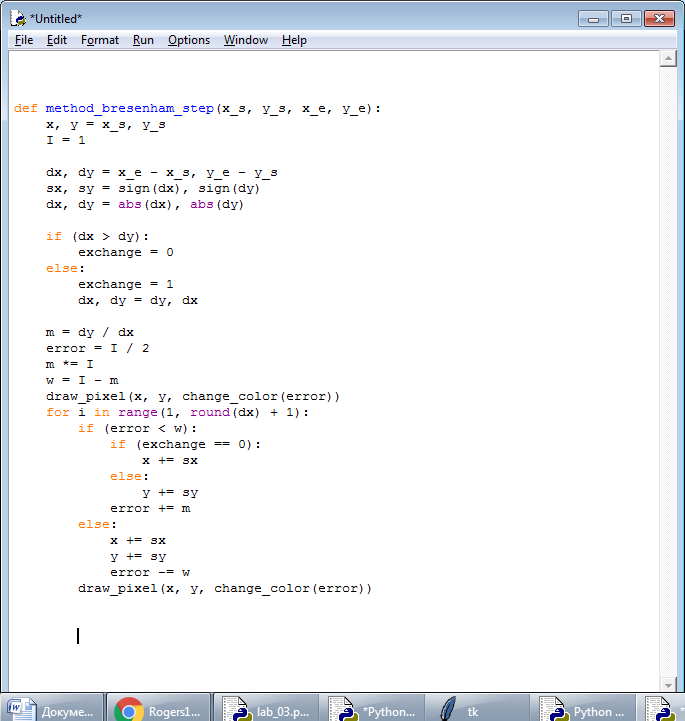
По сравнению с предыдущими алгоритмами немного изменились длина некоторых ступенек.

1. Алгоритм Брезенхема с устранением ступенчатости

Основная идея этого алгоритма состоит в том, чтобы сделать визуально ступеньки слабозаметными или практически незаметными. Пиксел рассматривается не как математическая точка, а как некоторая конечная область.

При наличии нескольких оттенков цвета внешний вид отрезка улучшается путем размывания его краев. Для этого интенсивность пиксела устанавливается пропорционально площади той части, которая находится под отрезком. Поэтому используются выражения для вычисления этой площади.

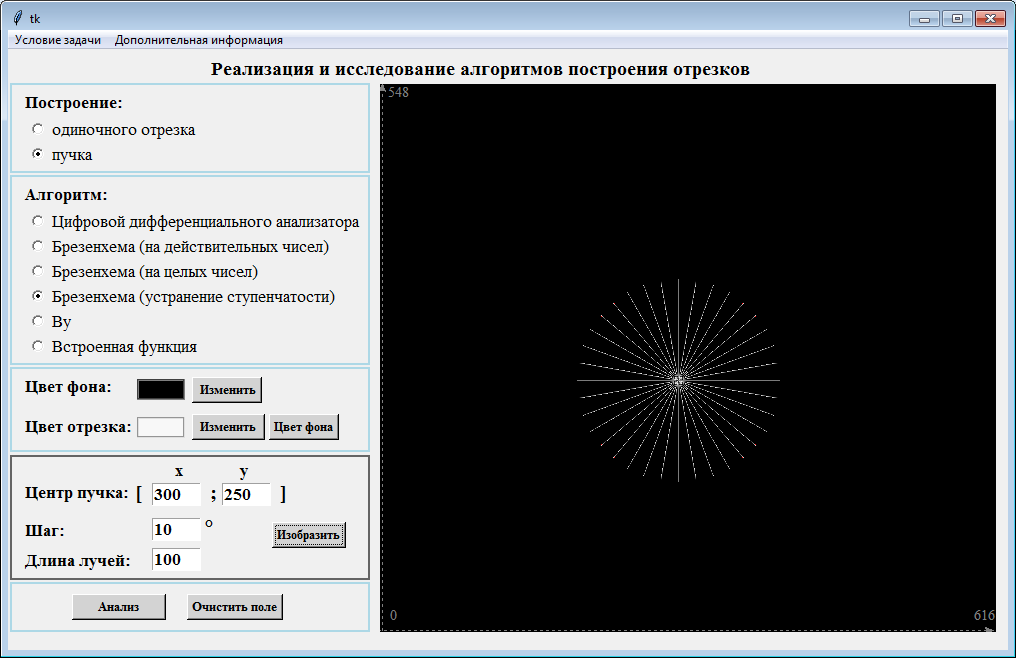
*Программная реализация:*

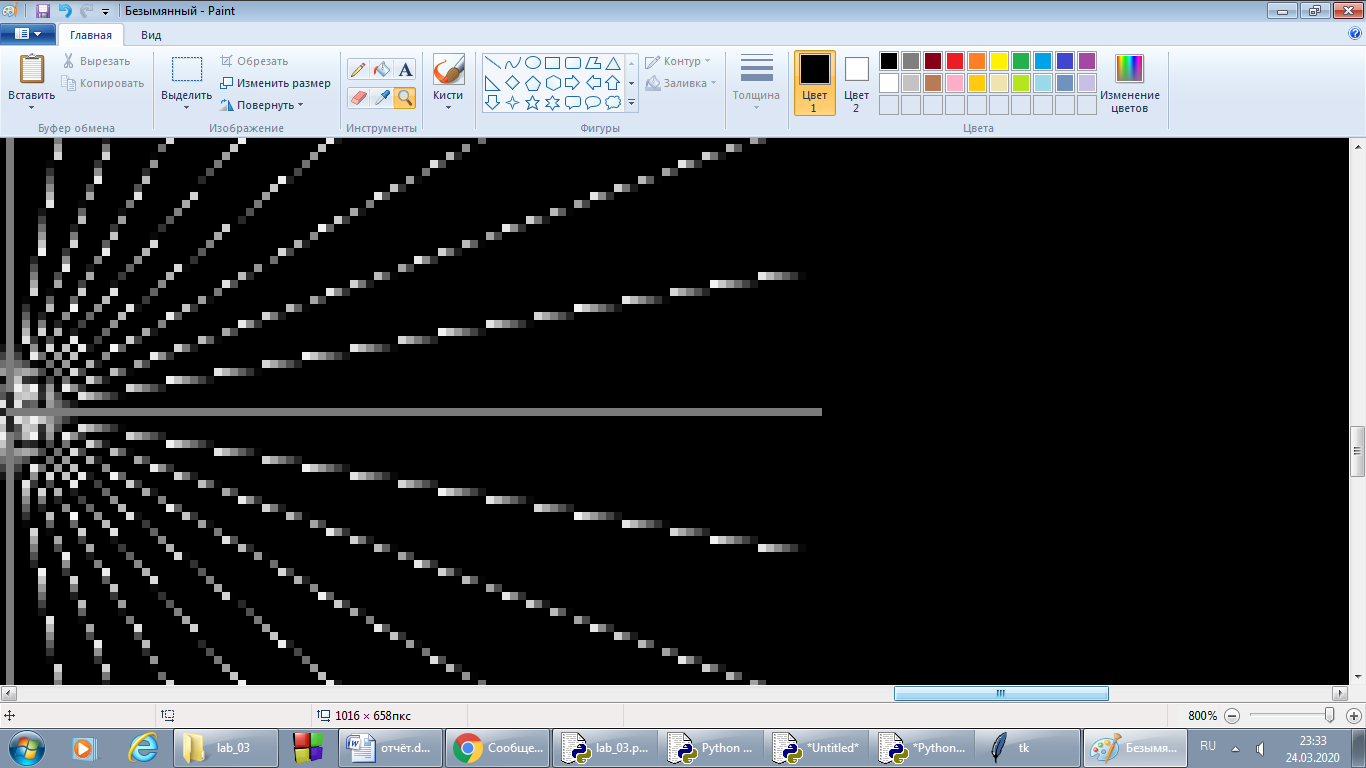


*Минусы данного алгоритма:*

1. более продолжительный по времени нежели предыдущие алгоритмы
2. если сравнивать с другими алгоритмами сглаживания, то у этого алгоритма визуальные характеристики хуже (по сравнению, например, с Ву)

Но, несмотря на минусы, этот алгоритм полезен для рисования многоугольников (при закрашивании внутренней часть ступени).

*Пример работы программы:*

Увеличение:

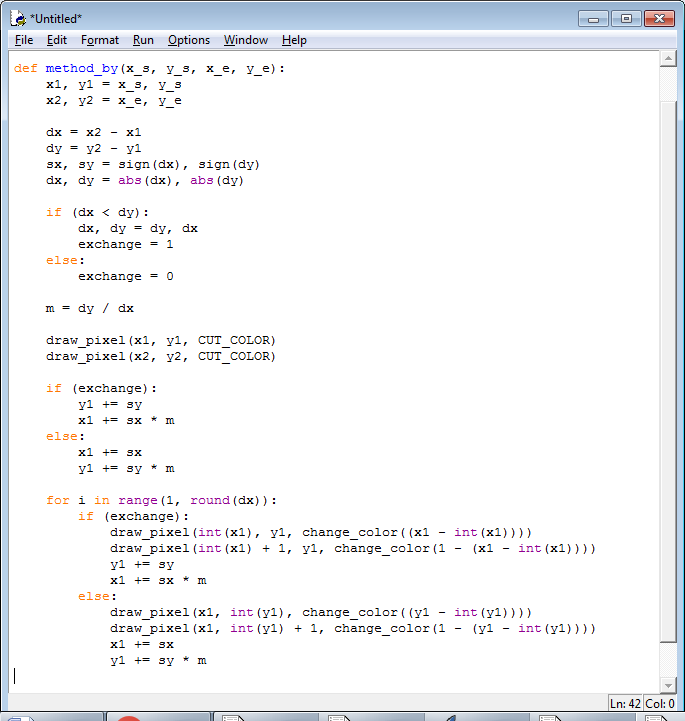
Можно наблюдать градиент цвета, за счёт которого и происходит размытие границ.

1. Алгоритм Ву

Отрезок высвечивается толщиной в два пиксела, и их суммарная интенсивность на каждом шаге постоянна. Само сглаживание осуществляется за счёт перераспределения интенсивности между этими двумя пикселами.

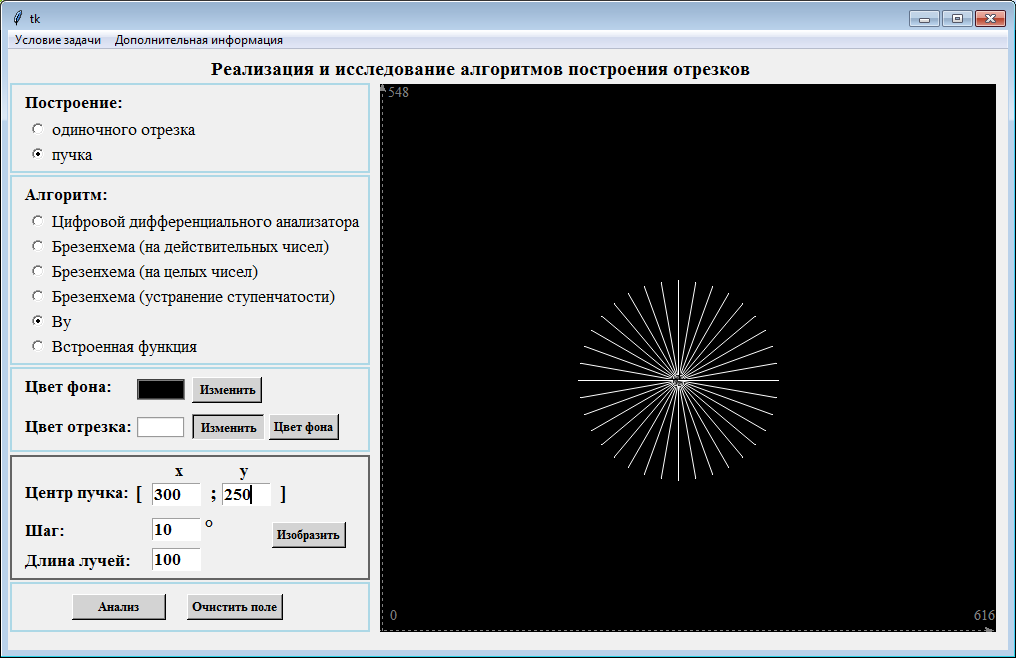
Интенсивность определяется в зависимости от расстояния между пикселом и точкой, расположенном на «идеальном» отрезке, чем ближе к точке, тем больше его интенсивность.

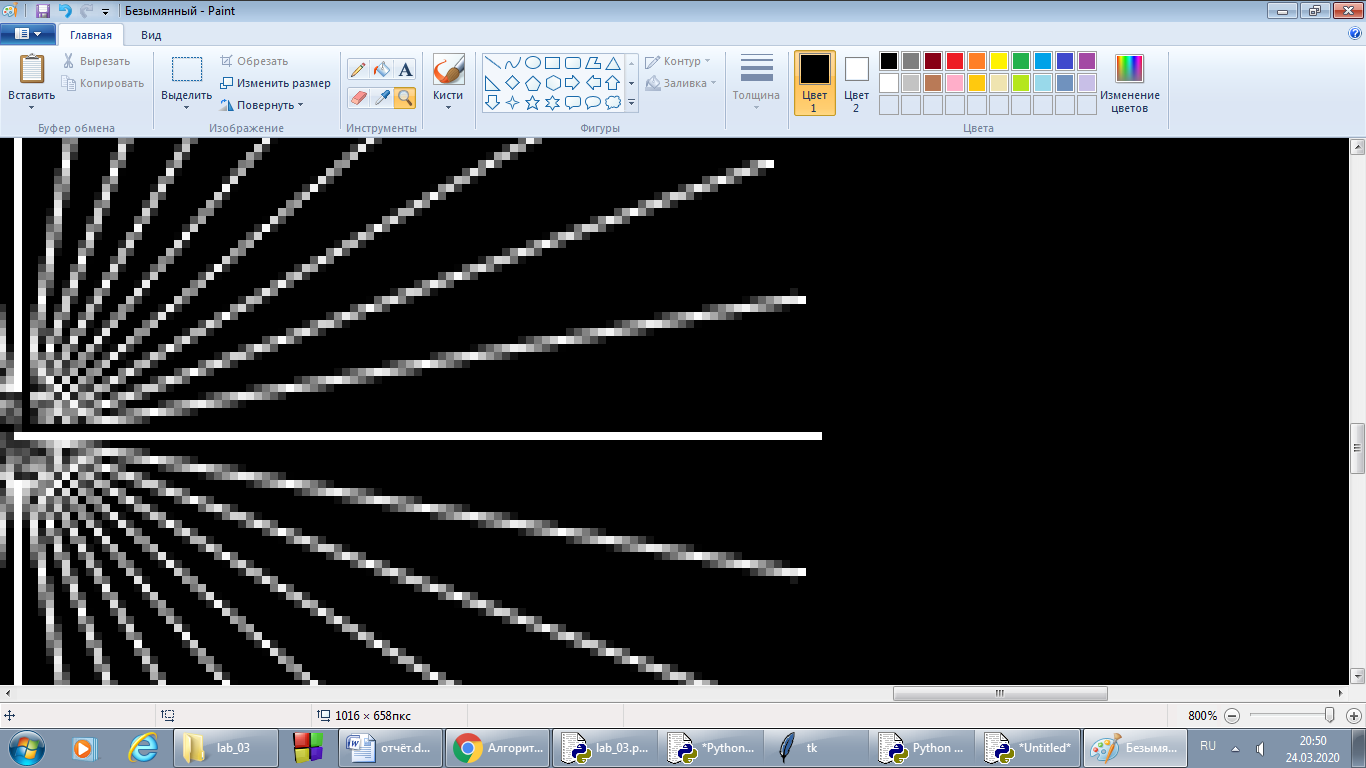
*Программная реализация:*



*Минусы данного алгоритма:*

1. продолжительный по времени

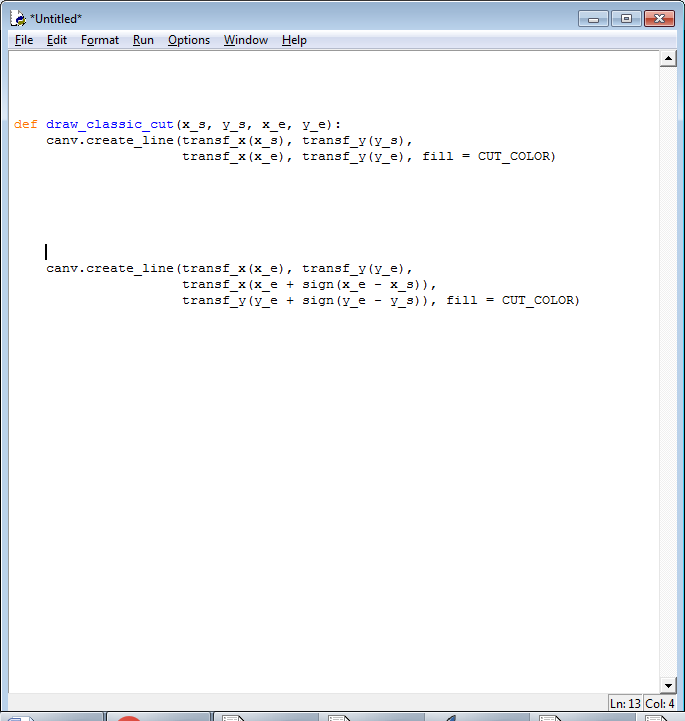
*Пример работы программы:*

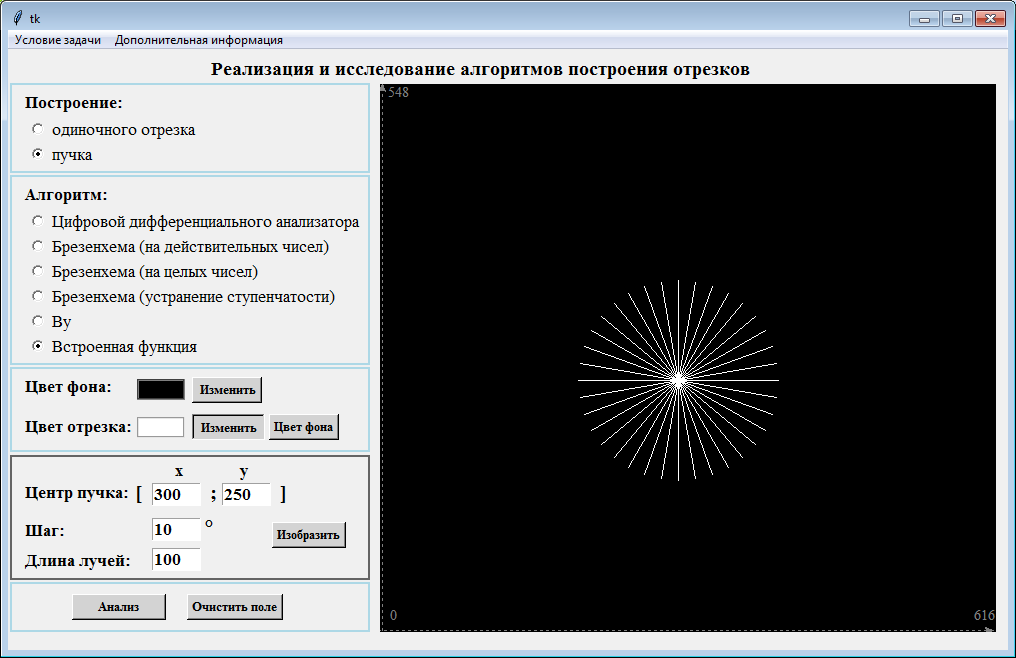
Увеличение:

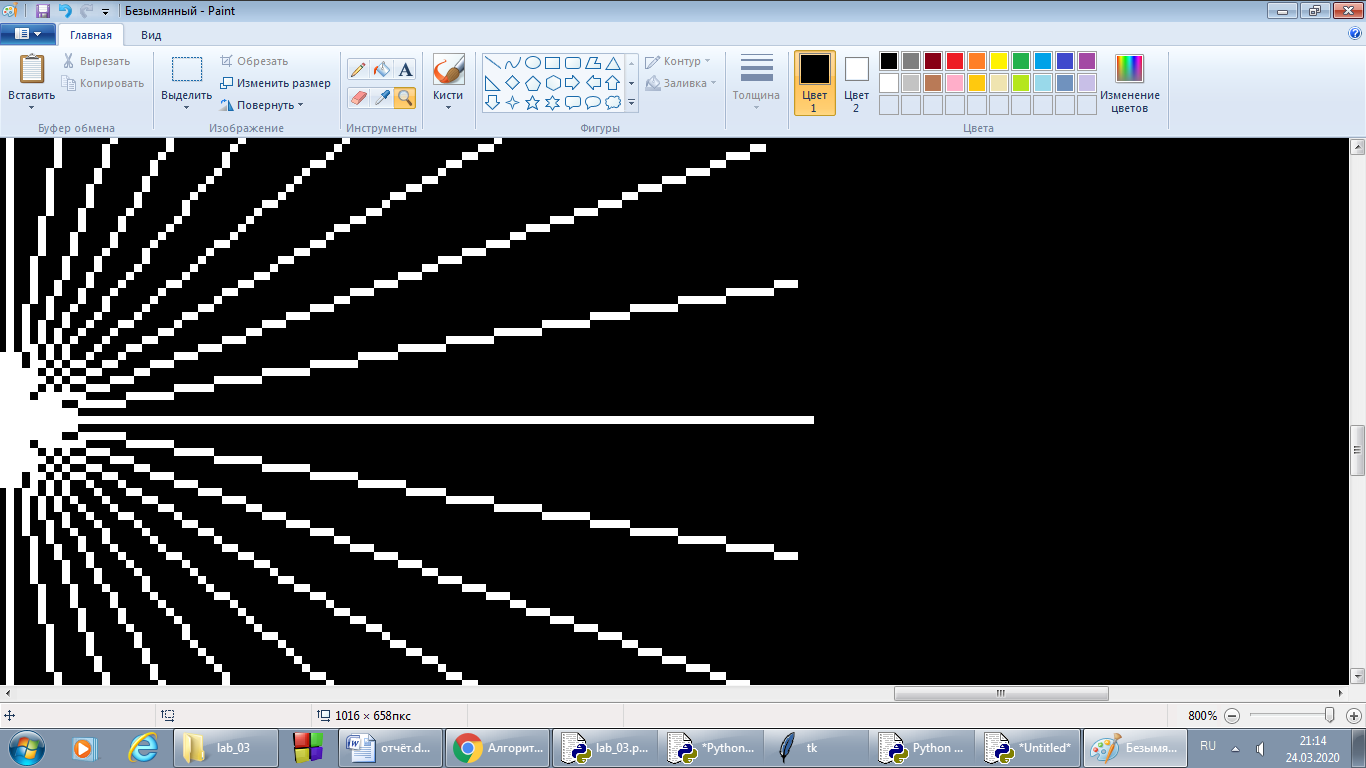
Четко видны визуальные преимущества этого алгоритма по сравнению со всеми предыдущими (сглаживание выглядит более правдоподобно).

1. Встроенная функция

Вызов соответствующей функции:



*Пример работы программы:*

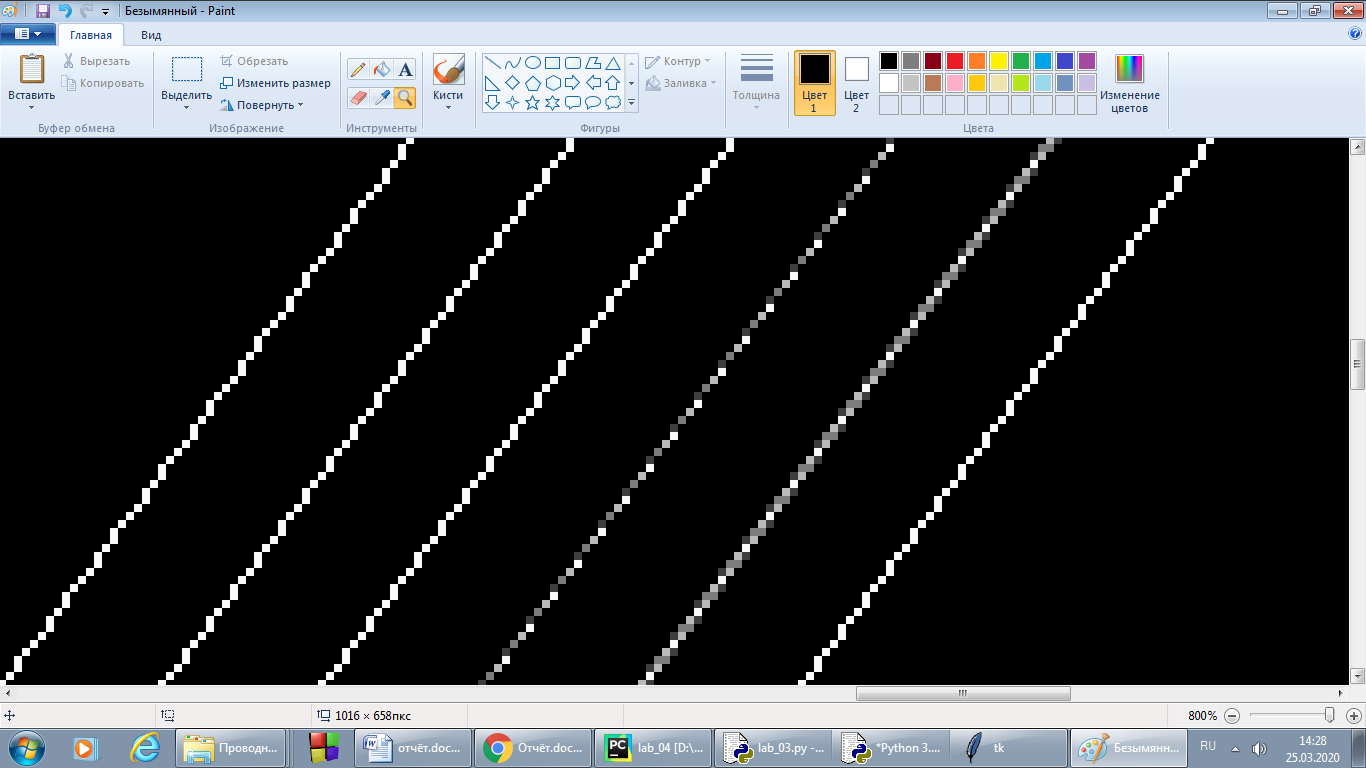
Увеличение:

У встроенной функции нет устранения ступенчатости.

**Исследование временных и визуальных характеристик**

Произведём сравнение визуальных характеристик путём изображения одного и того же объекта разными алгоритмами (обозначаются соответственно разными цветами).

Слева направо изображены алгоритмы ЦДА, Брезенхема (вещественный, целый и с устранением ступенчатости), Ву и встроенный метод рисования отрезка графического интерфейса.



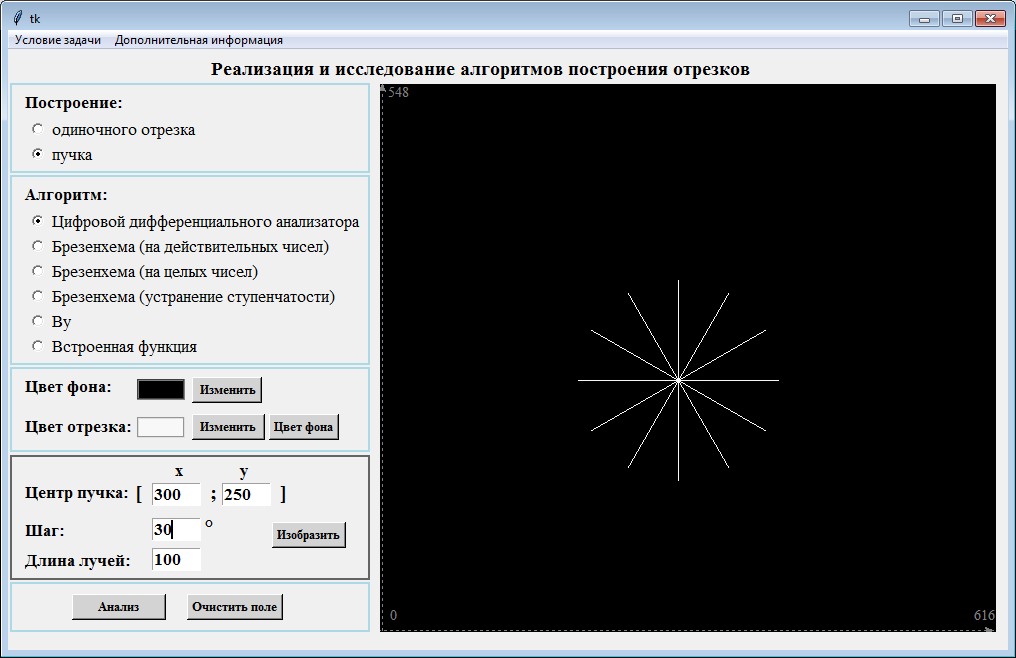
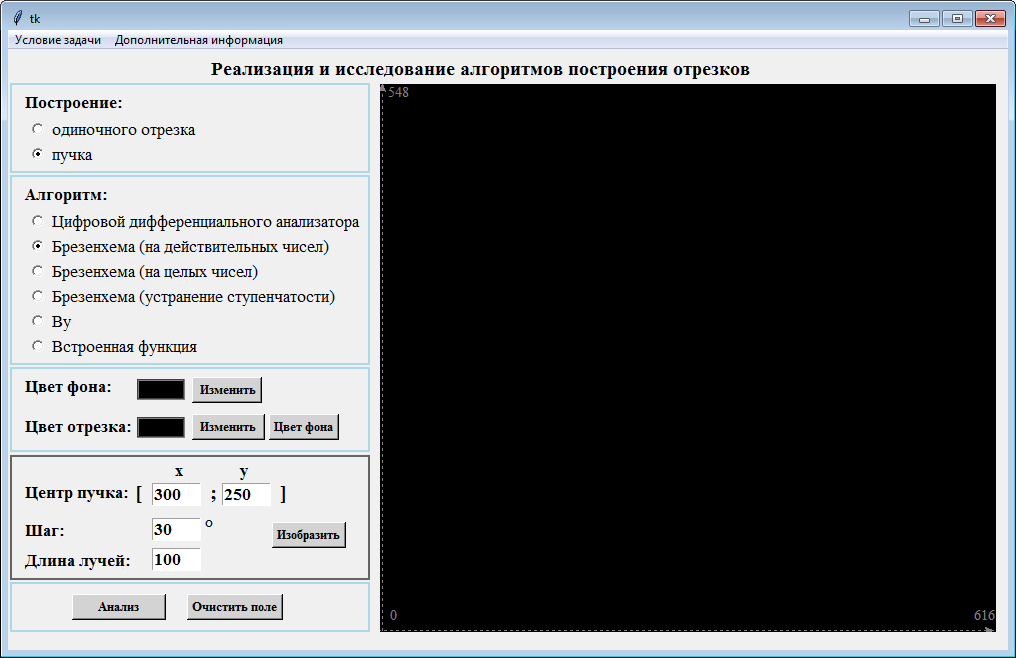
Благодаря увеличению видно, что ЦДА и Брезенхем без сглаживания не имеют визуальных различий, что подтверждается и при наложении одного отрезка на другой, с разными цветами.

Также для сравнения был взят пучок лучей (чтобы учесть различные варианты угла наклона), длина которых 100, расположенных с шагом в 30 градусов.

Исходные данные для алгоритмов одинаковые (т.к. подразумевается наложение).

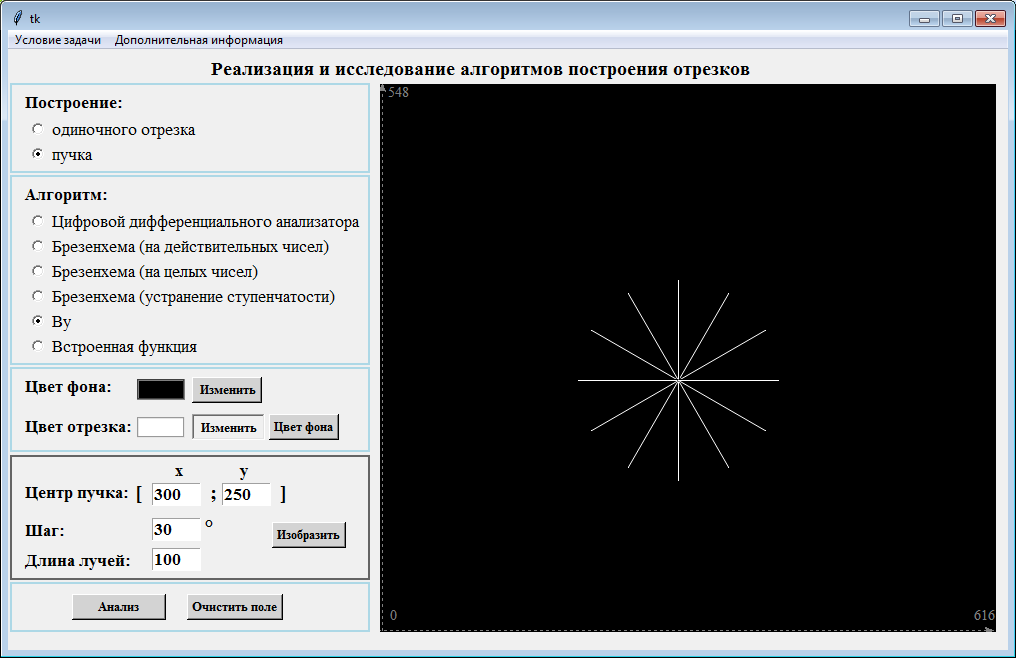
Цвет второго алгоритма, с которым происходило сравнение, чёрный (цвет фона, для того, чтобы разница была заметнее).

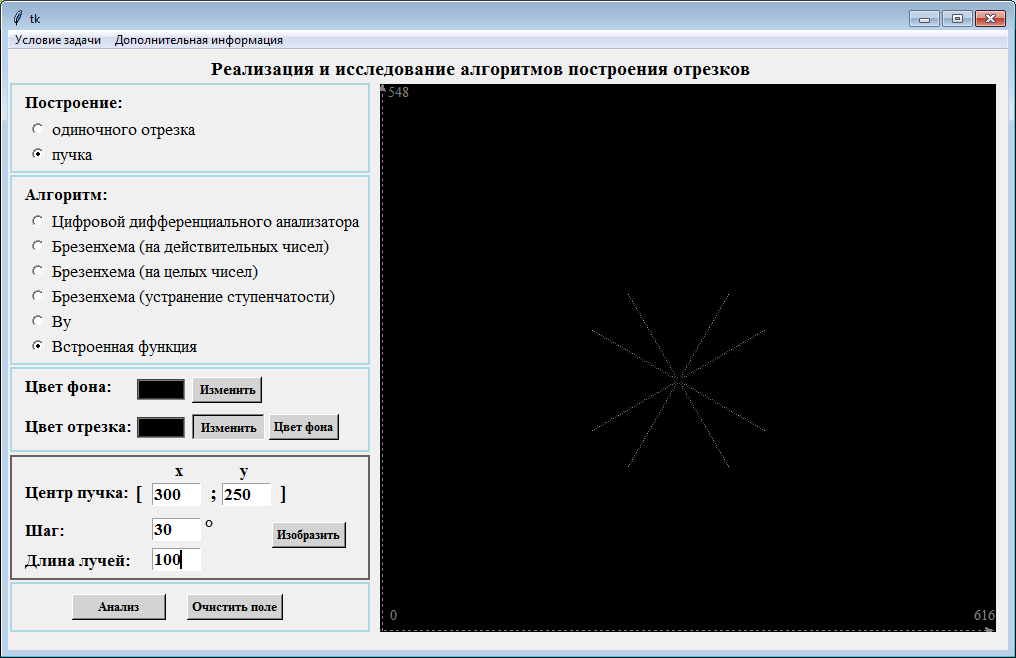
Если сравнивать алгоритм ЦДА и Брезенхема на действительных числах, то можно наблюдать такую ситуацию.

а) ЦДА (белый цвет) б) Брезенхем (чёрный)

То есть, второй алгоритм полностью закрасил первый (то же самое работает и в обратную сторону).

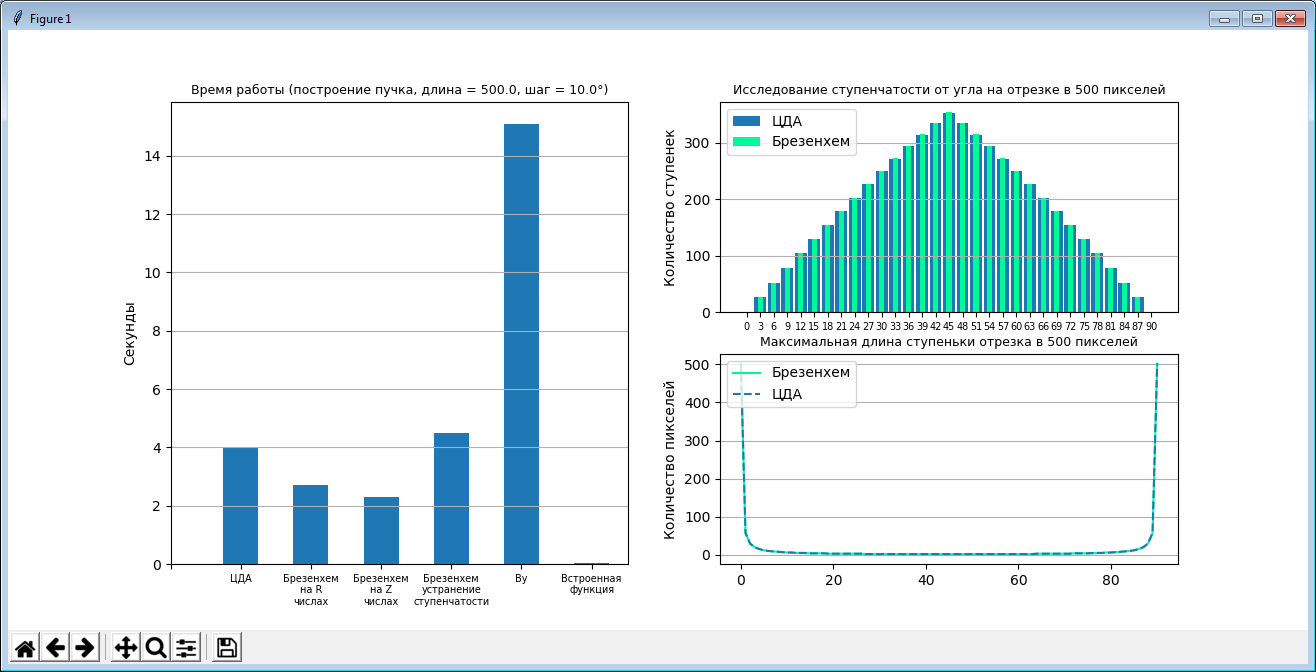
А если сравнить алгоритм Ву и встроенный, то возникает такая ситуация:

а) Ву (белый) б) встроенная(чёрный)

То есть, встроенная функция не смогла полностью закрасить пучок лучей, построенный алгоритмом Ву (в силу особенностей самого алгоритма). Похожая ситуация будет наблюдаться при попытке закрасить пучок, нарисованным по алгоритму Брезенхема с устранением ступенчатости, встроенной функцией.

Ниже представлены диаграммы анализа:

1. времени анализа работы (построения пучка, длина лучей которого 500, шаг 10 градусов)
2. ступенчатости в зависимости от угла (на отрезке в 500 пикселей)
3. максимальная длина ступеньки в зависимости от угла (на отрезке в 500 пикселей)



По первой диаграмме можно «отсортировать» алгоритмы от самого быстрого к самому медленному:

1. Встроенный
2. Брезенхем (на целых числах)
3. Брезенхем (на вещественных)
4. ЦДА
5. Брезенхем (с устранением ступенчатости)
6. Ву

По второй и третьей диаграмме можно сделать вывод о том, что, чем дальше расположен отрезок от горизонтали и вертикали, тем короче его ступеньки и тем больше их количество. Кроме того, ЦДА и Брезенхем (будь то целочисленный или вещественный) не имеют визуальной разницы, это заключение можно сделать из того, что их графики ступенчатости практически совпадают.

**Заключение**

Таким образом, в ходе лабораторной работы были изучены и реализованы несколько алгоритмов построения отрезков, проанализирована их работа, и проведён сравнительный анализ, по итогам которого были сделаны соответствующие выводы.