|  |  |
| --- | --- |
| 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 *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

по дисциплине:

Разработка САПР

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Группа |  | РК6-11М, РК6-12М |
| Тип задания |  | Лабораторная работа |
| Тема лабораторной работы |  | Реализация геометрического решателя |

Выполнили:

**Абидоков Р. Ш.**

**Пузаков Г. К.**

*Москва, 2020 г.*

Оглавление

[Задание на лабораторную работу 3](#_Toc60683520)

[Теоретическая часть 3](#_Toc60683521)

[Интерфейс программы 6](#_Toc60683522)

[Примеры наложения ограничений 7](#_Toc60683523)

[Описание программной реализации 8](#_Toc60683524)

[Присутствующие недочеты 10](#_Toc60683525)

[Заключение 11](#_Toc60683526)

# Задание на лабораторную работу

Реализовать графический редактор с возможностью создания точек и отрезков, наложения геометрических ограничений:

* Совпадения двух точек
* Расстояния между 2 точками
* Параллельности двух прямых
* Перпендикулярности двух прямых
* Угла между двумя прямыми
* Горизонтальности прямой
* Вертикальности прямой
* Принадлежности точки прямой

И возможностью расчета положения геометрии с учетом удовлетворения заданным ограничениям на основе принципа наименьших изменений.

# Теоретическая часть

С математической точки зрения удовлетворение геометрическим ограничениям означает нахождение такого изменения координат точек (т.к. отрезок также описывается двумя точками), при котором удовлетворяется система уравнений

являющаяся выражением наложенных ограничений, где n – количество точек, m – количество ограничений. Данная система уравнений в общем случае недоопределена, поскольку количество уравнений может быть меньше, чем 2n неизвестных.

Для того, чтобы сделать возможным нахождение однозначного решения, используется принцип наименьших изменений. Дополнительно к наложенным геометрическим ограничениям добавляется условие минимизации функции

Таким образом исходная задача приводится к задаче оптимизации с ограничениями вида равенств и решается помощью метода множителей Лагранжа. Составляется функция Лагранжа вида:

И разрешающая система уравнений

Таким образом, количество уравнений совпадает с количеством неизвестных –

и тех, и других 2n+m штук. С учетом вида функций и :

Получаем итоговую систему:

Выразим в явном виде функции ограничений (производные не приводятся для компактности).

* Расстояние между точками, и, как частный случай, их совпадение:
* Параллельность двух прямых (нулевое векторное произведение):
* Перпендикулярность двух прямых (нулевое скалярное произведение):
* Угол между двумя прямыми
* Горизонтальность прямой
* Вертикальность прямой
* Принадлежность точки прямой

# Интерфейс программы

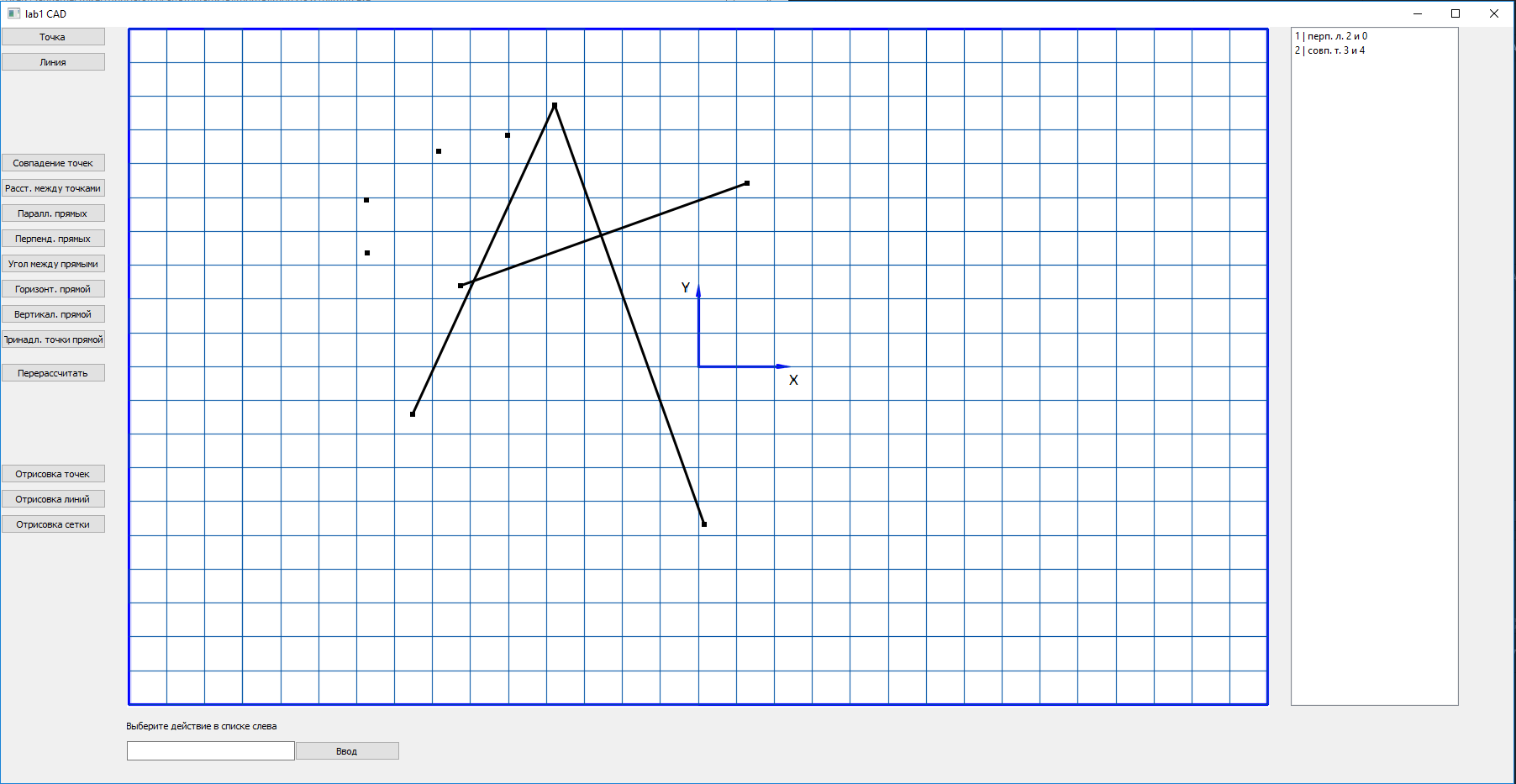


Рис. 1

Окно программы можно разделить на четыре зоны. В центре находится поле, внутри которого происходит построение графики. Справа находится лист, в котором выведен список наложенных геометрических ограничений.

Слева расположена панель с кнопками. Две верхних – "точка" и "линия" – отвечают, соответственно, за построение точки и линии. Ниже расположены кнопки наложения геометрических ограничений, под ними кнопка "перерасчитать", производящая повторное решение системы уравнений. Три отдельных кнопки ниже отвечают за включение-выключение отрисовки точек, линий и сетки поля.

Внизу текстовая строка, выводящая сообщения, и поле ввода с кнопкой, применяемое для ограничений расстояния между точками и угла между прямыми.

# Примеры наложения ограничений



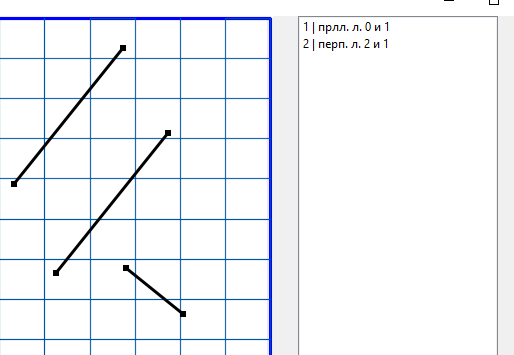


Рис. 2 Параллельность и перпендикулярность

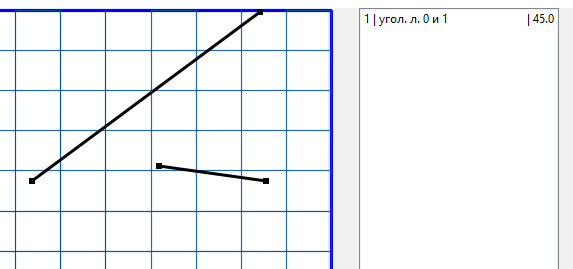
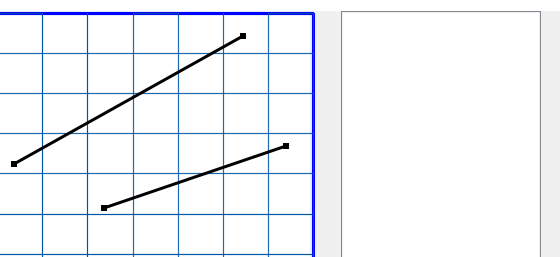


Рис. 3 Угол между прямыми

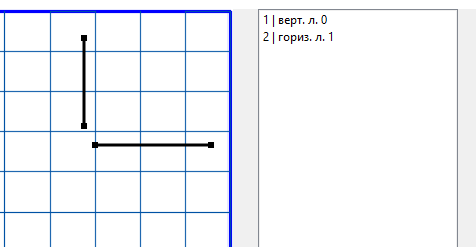
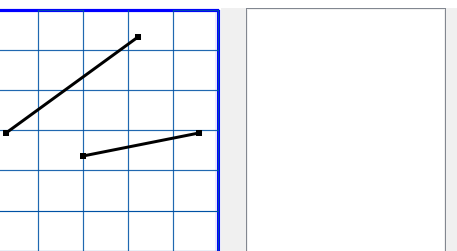


Рис. 4 Горизонтальность и вертикальность прямой

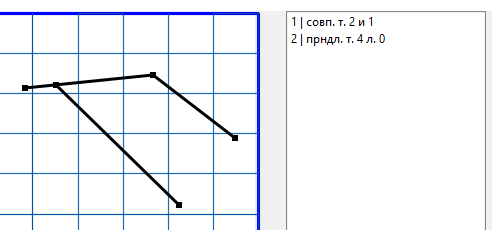
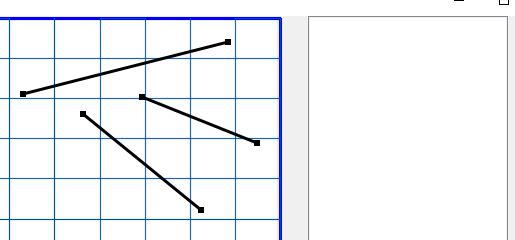


Рис. 5 Совпадения точек и принадлежность точки прямой

# Описание программной реализации

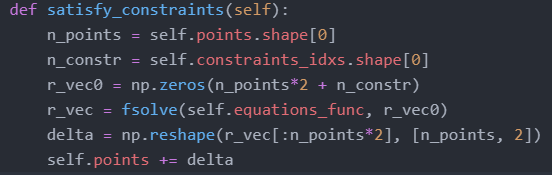
Программа написана на языке Python 3.7.2 с использованием библиотеки PyQt5, отвечающей за графическую составляющую, и библиотек Numpy и Scipy для удобной работы с массивами и решения системы нелинейных уравнений.

С точки зрения архитектуры произведено разделение на три класса. Класс GuiClass непосредственно вызывается в main и обеспечивает отрисовку интерфейса, обработку событий, таких как щелчки мыши или нажатия кнопок, а также взаимодействие с классами StateClass и GeometryClass, экземпляры которых создаются в качестве полей.

StateClass отвечает за последовательную корректную смену состояний программы и вывод соответствующих сообщений пользователю, GeometryClass – за хранение координат точек, геометрических ограничений, формирование системы нелинейных уравнений и ее решение.

Удовлетворение граничным условием производится посредством вызова функции satisfy\_constraints, приведенной в Листинге 1.

Листинг 1



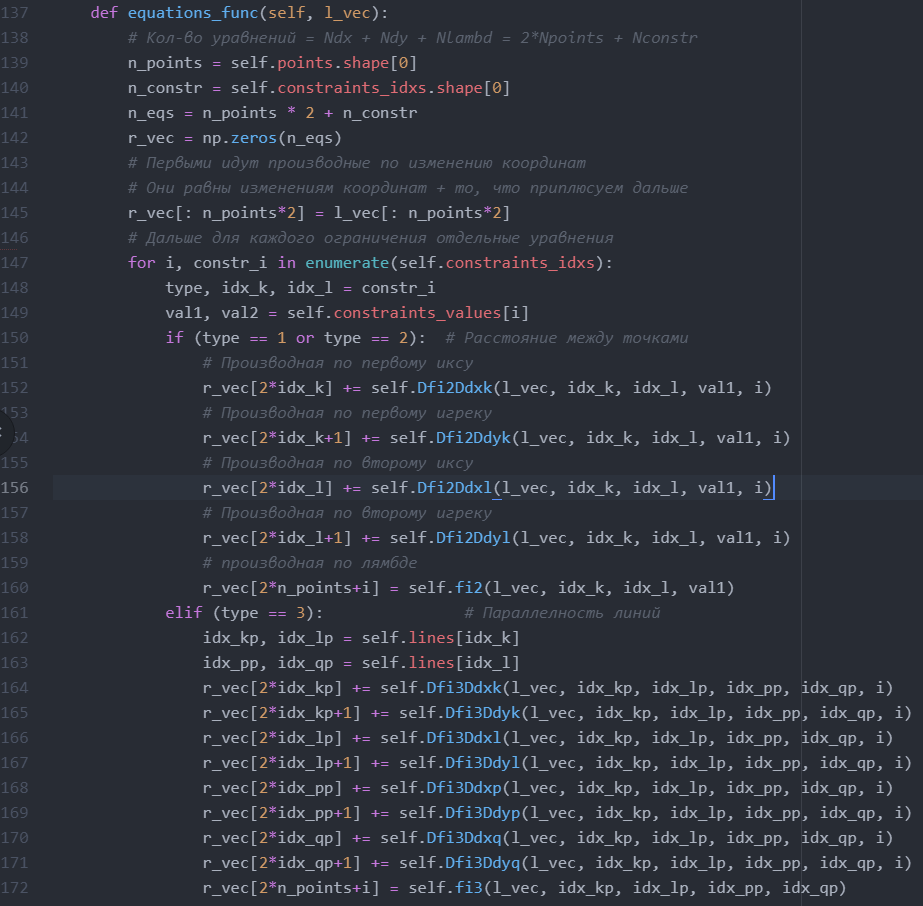
Искомый вектор r\_vec размерности 2n+m имеет следующую структуру: первые 2n элементов – – т.е. изменения координат точек, оставшиеся m – неизвестные множители Лагранжа .

Формируется вектор начальных приближений r\_vec0 (нулевых) и вместе с функцией equations\_func, реализующей систему уравнений (т.е. по предложенному r\_vec формирующей выходной вектор) передается функции fsolve модуля scipy.optimize, которая непосредственно ищет корни системы. Далее координаты точек изменяются на найденные величины (т.е. на первые 2n элемента r\_vec).

Сама функция equations\_func работает следующим образом: осуществляется проход по таблице, содержащей ограничения в виде столбцов "тип-индекс1-индекс2-величина1-величина2", где в зависимости от типа индексы могут соответствовать номерам точек либо линий, для каждого ограничения в выходном векторе увеличивается значения производных по тем изменениям координат, которые участвуют в этом ограничении, а также производной по соответствующему множителю Лагранжа. Часть кода функции (для ограничений расстояния между точками и параллельности прямых) приведена в Листинге 2.

Полный исходный код программы находится в приложенных файлах и не включен в отчет в силу его громоздкости (порядка 1200 строк).

Листинг 2



# Присутствующие недочеты

1. Отсутствует возможность изменения масштаба и движения системы координат
2. После расчета нового положения точек, часть из них может оказаться вне поля, ограниченного рамкой. В этом случае графическое отображение будет не совсем корректным (Рис. 6)

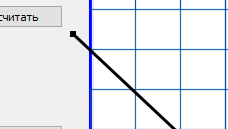


Рис. 6 (выход линии за границу рамки)

1. Если выбрать ограничение в списке справа и таким образом активировать подсветку точек/линий, а затем отключить отображение соответствующих элементов, то подсвеченная геометрия продолжит отображаться

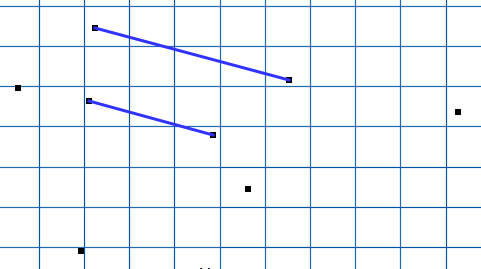
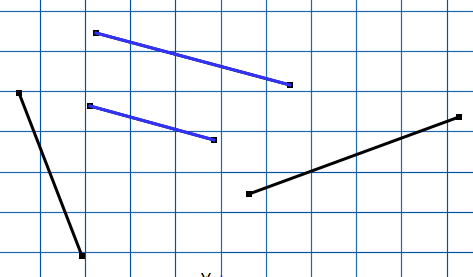


Рис. 7 (при выключении отображения линий)

1. При наложении ограничения на расстояние между точками, отличное от нулевого, поведение может быть необычным – например, в случае, когда оптимальным решением являлось бы сближение точек вдоль линии, их соединяющей, может произойти поворот, как это показано на Рис. 8. Возможным объяснением такого поведения могут служить проблемы решателя при поиске корней уравнения.

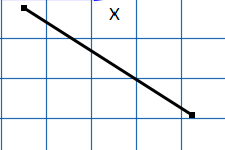
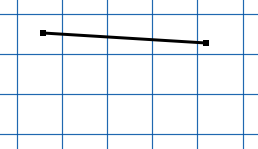


Рис. 8

# Заключение

Был реализован графический редактор с возможностью построения двухмерной геометрии, состоящей из точек и отрезков и наложения геометрических ограничений. При реализации использован язык программирования Python, графическая библиотека PyQt5 и математические библиотеки Numpy, Pandas. Суммарный объем исходного кода составил примерно 1200 строк.

Для указанных выше недочетов могут быть предложены следующие решения:

1. *Масштаб и движение системы координат* – разделение системы координат, использующейся при отрисовке, и абсолютной системы координат, в которой записаны положения точек, непосредственный пересчет одной в другую при отрисовке с помощью матриц перехода
2. *Выход части объектов за границу поля* – выполнять проверку на принадлежность полю, не отрисовывать объекты, целиком находящие вне поля, для линии, пересекающей рамку, конечную точку заменять на точку пересечения линии и границы
3. *Продолжение подсветки элементов, на которые наложены ограничения* – добавить проверку флага при отрисовке, аналогичную той, что присутствует для неподсвеченных линий
4. *Поворот при наложении ограничения расстояния* – может помочь смена решателя и/или использование начальных приближений, отличных от нулевых.