

Санкт-Петербургский Политехнический университет Петра Великого  
Институт прикладной математики и механики  
**Высшая школа прикладной математики и вычислительной физики**

## **Отчёт по научно-исследовательской работе**

на тему:

Визуализатор данных и ответов в вычислительной геометрии  
по направлению 01.04.02 'Прикладная математика и информатика'  
(профиль 01.04.02\_02 Математические методы анализа и визуализации  
данных)

Студент группы 3640102/00201 \_\_\_\_\_ Н.В.Лансков

Студент группы 3640102/00201 \_\_\_\_\_ М.А.Нахатович

Оценка научного руководителя: \_\_\_\_\_ С.Э.Володарский

Санкт-Петербург  
2021 г

# Содержание

<b>1</b>	<b>Введение</b>	<b>2</b>
<b>2</b>	<b>Постановка задачи</b>	<b>2</b>
<b>3</b>	<b>Анализ существующих решений</b>	<b>2</b>
<b>4</b>	<b>Описание проделанной работы</b>	<b>2</b>
4.1	Текстовый формат . . . . .	2
4.2	Приложение . . . . .	3
<b>5</b>	<b>Результаты</b>	<b>4</b>
<b>6</b>	<b>Описание дальнейшей работы</b>	<b>5</b>
<b>7</b>	<b>Список иллюстраций</b>	<b>6</b>
<b>8</b>	<b>Список литературы</b>	<b>7</b>
<b>A</b>	<b>Список задач</b>	<b>8</b>

# 1 Введение

Для решения задач вычислительной геометрии разрабатываются специальные программы. Такие программы получают данные и возвращают результаты в виде текстовых файлов. Глядя на столбцы чисел, бывает трудно понять, корректны ли входные данные и верно ли решение, которое выдала программа. Проверки корректности входных и выходных данных могут быть реализованы в отдельной программе, но для каждой задачи нужна будет отдельная программа. В первоначальной отладке полезен инструмент, который отображает входные данные и решение. Целью данной работы является создание универсального инструмента для визуализации данных. Этот инструмент должен читать данные и ответы для широкого списка задач и отображать их в графическом виде.

## 2 Постановка задачи

Требуется создать универсальный текстовый формат для описания данных. Этот формат должен поддерживать входные и выходные данные задач из списка. Полный список задач приведён в приложении А. Также требуется разработать приложение для визуализации данных, описанных с применением разработанного формата, на языке JavaScript. Приложение должно работать в браузере и поддерживать отрисовку больших объёмов данных. На вход программа получает текстовый файл с данными, которые требуется отрисовать. Описание данных должно соответствовать разработанному формату.

## 3 Анализ существующих решений

Самым известным решением является программа **GeoGebra** [1].

GeoGebra — это бесплатная, кроссплатформенная динамическая математическая программа, включающая в себя геометрию, алгебру, таблицы, графы, статистику и арифметику, в одном пакете.

Однако, для наших целей она не подходит, так как обладает следующими недостатками:

1. Поддерживает описание объектов только в формате XML, который трудно читать
2. Не поддерживает отрисовку большого числа объектов (~100000).

Поэтому было решено разработать собственное решение для визуализации данных.

## 4 Описание проделанной работы

### 4.1 Текстовый формат

В основе формата для описания данных было решено использовать **JSON** формат. Данный выбор обусловлен следующими преимуществами **JSON**:

1. Встроенная поддержка во многих языках программирования, стандарт для **JavaScript**
2. Поддержка автоматической валидации при помощи **JSON Schema** [4]
3. Легко читается людьми

Формат имеет следующую единообразную структуру:

- Любой документ, поддерживающий разработанный формат, содержит два поля: объект **elements** и массив **visualizations**
- Внутри поля **elements** содержатся все объекты, которые нужны для визуализации данных. Значение каждого поля объекта **elements** может быть как непосредственно объектом визуализации (многоугольник, отрезок, точка и т.д.), так и списком таких объектов. Списки могут быть вложенными. Объекты визуализации содержат поле **type** с указанием типа объекта, а также другие поля, специфичные для указанного типа объекта. Кроме того, объекты также могут быть представлены ссылкой на другой элемент данных, описанный внутри **elements**
- Массив **visualizations** содержит объекты-ссылки на объекты данных, указанные в **elements**, которые требуется отрисовать

Таким образом, данный формат легко расширить новыми типами объектов.

## 4.2 Приложение

Для визуализации данных было разработано приложение на языке JavaScript. В качестве метода отрисовки был использован WebGL renderer, подключённый посредством библиотеки Three.js [5]. Возможности быстрого 'зумирования' и 'перемещения сцены' реализованы при помощи библиотеки D3.js[6].

Приложение состоит из трёх основных элементов интерфейса: меню, в котором есть возможность загрузить новый файл с данными для визуализации, блок информации, в котором отображается информация об отрисованных объектах, и непосредственно полотно для отрисовки графических данных.

Также, приложение вычисляет некоторые простейшие метрики объектов, такие как, например, площадь и периметр многоугольников, для упрощения анализа результатов визуализации.

## 5 Результаты

В качестве демонстрации работы приложения ниже приведены результаты визуализации некоторых задач (Рис. 1, Рис. 2, Рис. 3).

Доступна демо-версия программы[2], и исходный код находится в открытом доступе[3].

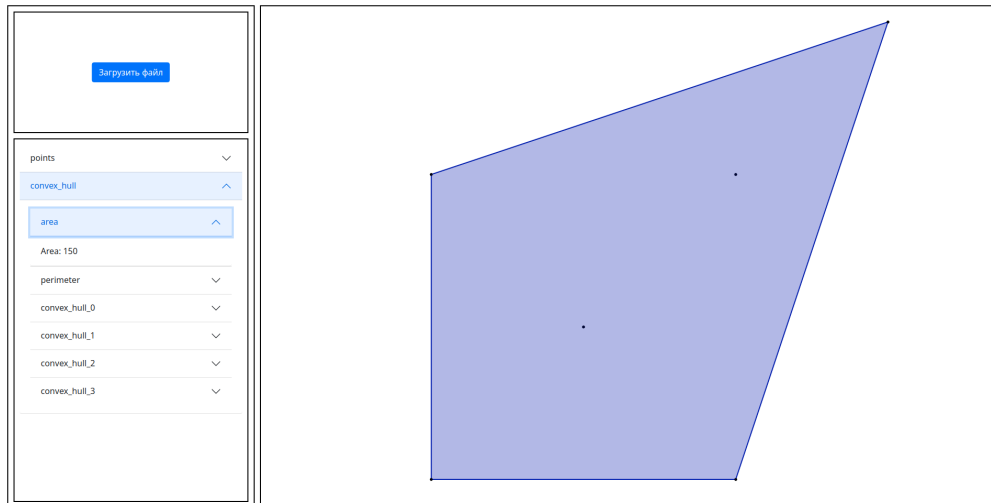


Рис. 1: Построение выпуклой оболочки для множества точек

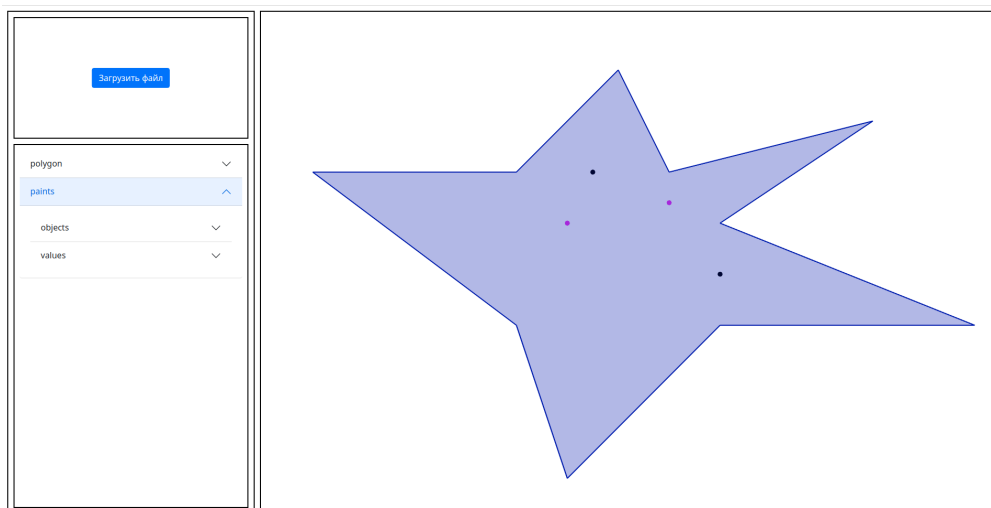


Рис. 2: Проверка звёздности многоугольника для заданных точек

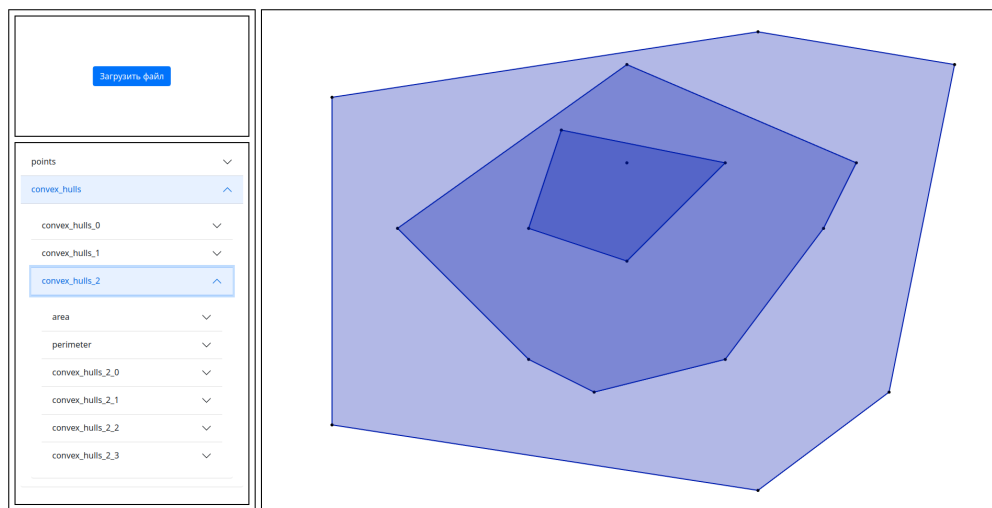


Рис. 3: Задача о чистке лука

## 6 Описание дальнейшей работы

В дальнейшем требуется доработать графический интерфейс программы, а также поддерживать возможность отрисовки большего числа объектов.

## 7 Список иллюстраций

1	Построение выпуклой оболочки для множества точек . . . . .	4
2	Проверка звёздности многоугольника для заданных точек . . . . .	4
3	Задача о чистке лука . . . . .	5

## 8 Список литературы

- [1] GeoGebra, <https://geogebra.org>.
- [2] Demo, <https://cg.leins275.xyz>.
- [3] Code, <https://github.com/PrimatElite/cg-visualizer>
- [4] JSON Schema, <https://json-schema.org/>
- [5] Three.js, <https://threejs.org/>
- [6] D3.js, <https://d3js.org/>



## А Список задач

1. Принадлежность точки многоугольнику
2. Диаметр выпуклого многоугольника
3. Выпуклая оболочка
4. Минимальная опорная прямая для множества точек
5. Ширина выпуклого многоугольника
6. Пересечение выпуклых многоугольников
7. Многоугольник на множестве вершин
8. Проверка монотонности многоугольника (направления заданы)
9. Проверка звёздности многоугольника (точки заданы)
10. Пересечение полуплоскостей
11. Проверка звёздности многоугольника (точка не задана)
12. Пересечение ортогональных семейств отрезков
13. Сумма Минковского двух выпуклых многоугольников
14. Минимальный описанный прямоугольник для выпуклого многоугольника
15. Диаграмма Вороного сторон выпуклого многоугольника
16. Чистка лука
17. Пересечение выпуклого многоугольника и прямой
18. Триангуляция многоугольника (любой алгоритм)
19. Экстремальная точка в выпуклом многоугольнике
20. Число протыканий триангуляции выпуклого многоугольника
21. Максимальный треугольник, вписанный в выпуклый многоугольник
22. Минимальный диск для множества точек
23. Диаграмма Вороного множества точек (любой алгоритм)
24. Проверка триангуляции многоугольника
25. Отыскание троек точек на одной прямой

26. Максимальный круг, вписанный в выпуклый многоугольник
27. Проверка простоты многоугольника
28. Расстояние между выпуклыми многоугольниками
29. Локализация точки в ППЛГ (любой метод)
30. Ближайшая пара на плоскости
31. Триангуляция Делоне (любой метод)
32. Пересечение отрезков
33. Пересечение ППЛГ
34. Пересечение окружностей
35. Восстановить множество точек по диаграмме Вороного