

Санкт-Петербургский государственный университет  
Математическое обеспечение и администрирование информационных систем  
Кафедра системного программирования

Черников Артем Александрович

# Создание приложения для построения и анализа кривых и поверхностей

Отчёт по учебной практике

Научный руководитель:  
к. т. н., доцент Т. А. Брыксин

Санкт-Петербург

2020

# Оглавление

<b>Введение .....</b>	<b>3</b>
<b>1. Постановка задачи .....</b>	<b>4</b>
<b>2. Обзор существующих решений .....</b>	<b>5</b>
2.1. Microsoft Mathematics .....	5
2.2. Advanced grapher .....	5
2.3. Rhinoceros .....	5
2.4. FairCurveModeler.....	5
2.5. Выводы.....	6
<b>3. Выбор инструментов .....</b>	<b>7</b>
3.1. Библиотека для графического интерфейса .....	7
<b>4. Архитектура приложения .....</b>	<b>8</b>
4.1. Синтаксический анализатор .....	9
4.2. Математические функции.....	9
4.3. Обработка событий.....	11
4.4. Графический интерфейс.....	12
<b>Заключение .....</b>	<b>14</b>
<b>Список литературы.....</b>	<b>15</b>

## Введение

Визуальное представление данных часто упрощает понимание устройства каких-либо сложных структур. В частности, это касается таких геометрических объектов, как поверхности и кривые. Часто работа с ними происходит аналитически, а при возникновении малейших трудностей возникает интуитивное желание изобразить эскиз рассматриваемого объекта, что нередко помогает при работе.

По этой причине нашли широкое применение приложения, позволяющие строить различные графики функций и проводить анализ над ними. Однако желающих использовать подобные приложения в контексте более углубленной геометрии гораздо меньше, чем тех, кому требуются простые незамысловатые инструменты. Таким образом, спрос на предложения, использующих дифференциальную геометрию, довольно мал. Это привело к тому, что выбор среди данных приложений не особо широк. Однако это не отменяет того, что некоторым людям всё же может потребоваться такого рода приложение.

Таким образом, было принято решение создать приложение, которое расширит выбор среди тех, которые способны содействовать при изучении дифференциальной геометрии.

# 1. Постановка задачи

Целью данной работы является создание приложения для построения кривых и поверхностей, а также их анализа с точки зрения дифференциальной геометрии. В реализации данного проекта принимало участие два человека.

Для достижения поставленной цели от автора работы требуется:

- реализовать систему распознавания математического выражения;
- разработать представление функций и систему операций над ними;
- разработать графический интерфейс приложения;
- реализовать систему обработки нажатий клавиш клавиатуры и кнопок мыши.

## 2. Обзор существующих решений

### 2.1. Microsoft Mathematics

Microsoft Mathematics<sup>1</sup> является бесплатной обучающей программой, которая позволяет пользователям решать математические и научные задачи. В первую очередь она предназначена для студентов как инструмент обучения. Может быть полезной при решении небольших задач по алгебре, геометрии, анализу, статистике, физике и другим дисциплинам, требующим математические расчеты.

Поддерживается построение поверхностей и кривых, заданных явно и параметрически, а также в полярных координатах.

Также в функции приложения входят решение уравнений, нахождения данных о треугольнике, а также средство преобразования единиц измерений. Решение сопровождается пошаговым объяснением.

### 2.2. Advanced grapher

Advanced grapher<sup>2</sup> – программа для построения графиков и их анализа.

Поддерживается построение двумерных и трехмерных графиков функций, заданных в полярных координатах, параметрически, через таблицы, неявно и через неравенства.

Приложение позволяет также находить приближенные значения корней функции, точки экстремума, выполнять дифференцирование и численное интегрирование.

### 2.3. Rhinoceros

Rhinoceros<sup>3</sup> – программное обеспечение для трехмерного моделирования.

Для кривых реализовано вычитывание длины, угла между кривыми, отображение базиса Френе. Для поверхностей – площади, главных кривизн и, соответственно, Гауссовой и средней кривизн.

### 2.4. FairCurveModeler

FairCurveModeler<sup>4</sup> предназначен для точного моделирования кривых и поверхностей. Также используется дифференциальная геометрия для «улучшения» кривых или поверхностей.

---

<sup>1</sup> <https://www.microsoft.com/ru-ru/download/details.aspx?id=15702>

<sup>2</sup> <https://www.alentum.com/agraper/>

<sup>3</sup> <https://www.rhino-3d.ru/>

<sup>4</sup> <http://spliner.ru/>

## 2.5. Выводы

Авторами данного проекта был проведен анализ существующих популярных приложений, максимально приближенных к реализации поставленной авторами проекта задачи. От приложения, которое планировалось создать, ожидалось, что оно и подойдет для обучения, и будет использовать углубленную геометрию. В результате анализа было выяснено, что существующие приложения либо отвечают преимущественно первому требованию, но используют базовую математику, такие как Microsoft Mathematics и Advanced grapher, либо преимущественно второму, но недостаточно хорошо подходят для обучения, а направлены скорее на решение специфических задач моделирования, такие как FairCurveModeler и Rhinoceros. Таким образом, было решено создать приложение, которое в полной мере отвечает обоим требованиям.

## **3. Выбор инструментов**

Для реализации проекта был выбран язык Java. Мощные существующие движки подключать было нецелесообразно, поэтому были использованы стандартные инструкции рендеринга в OpenGL, так как ушло бы много времени на изучение новых графически библиотек. С OpenGL же уже был опыт работы у обоих участников проекта.

### **3.1. Библиотека для графического интерфейса**

В качестве библиотеки, предоставляющей возможность создавать графический пользовательский интерфейс, была выбрана LWJGUI [4], так как она создана как альтернатива JavaFX, с которой был опыт работы у обоих участников проекта, что облегчило ее усвоение. Также эта библиотека предоставляет больший по сравнению с другими набор элементов интерфейса.

## 4. Архитектура приложения

На рис. 1 представлена общая архитектура приложения. Голубым цветом выделены пакеты, которые реализовал автор работы, зелёным – те пакеты, над которыми работали оба человека, реализующие проект.

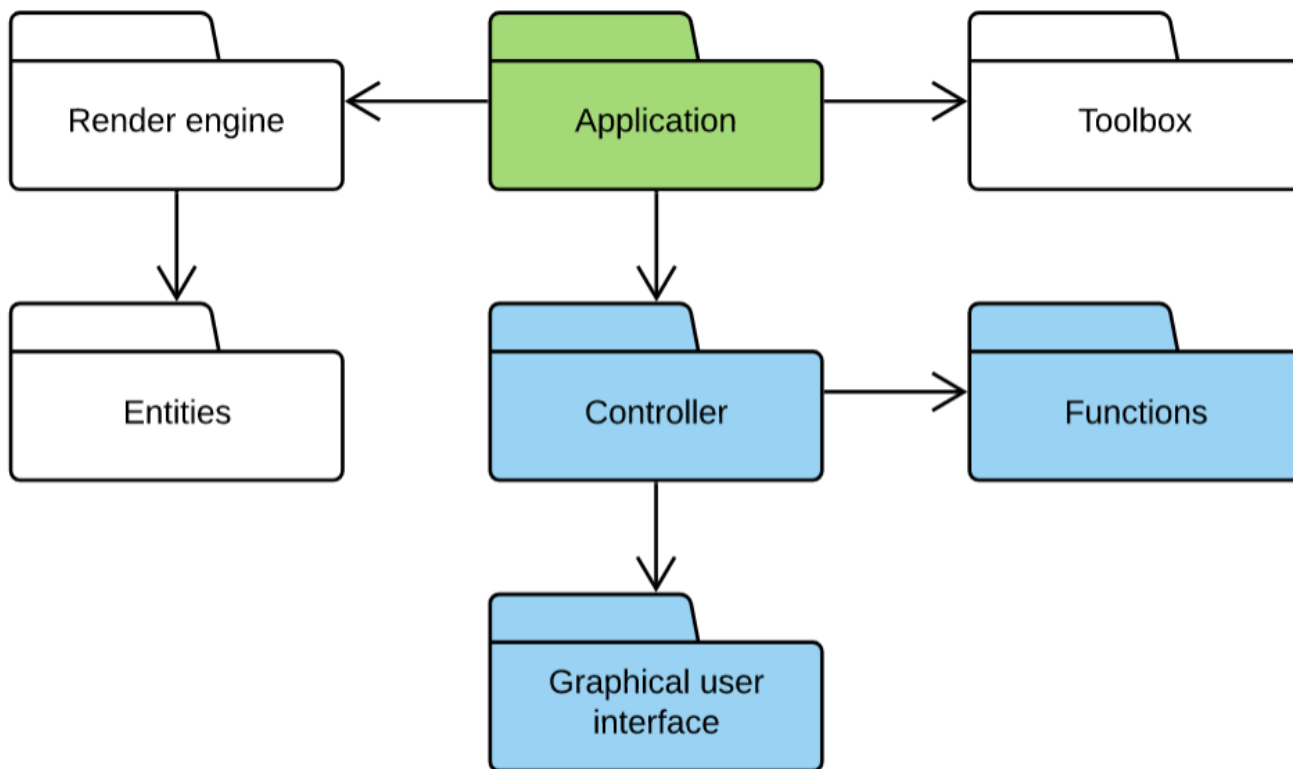


Рис. 1: Общая архитектура приложения

В пакете «Render engine» содержатся все классы, представляющие в совокупности инструменты вычисления и отображения визуальной информации в контексте данного приложения. В нём реализованы все инструменты, обрабатывающие поверхности, кривые, вектора и освещение.

Вышеупомянутый пакет использует объекты классов из пакета «Entities» для отображения их на экране. В пакете Entities содержатся классы, представляющие плоскость, кривую, вектор, а также камеру и свет.

Пакет «Toolbox» содержит в себе вспомогательные инструменты для удобной работы с данными.

Пакет «Controller» отвечает за обработку нажатия клавиш мыши и кнопок на клавиатуре.

В пакете «Functions» содержатся инструменты для представления функций, составления функции по строковому представлению и проведения операций над функциями.



Пакет «Graphical User Interface» представляет собой набор утилит для представления элементов графического интерфейса.

## 4.1. Синтаксический анализатор

За синтаксический анализ математического выражения отвечает класс `ExpressionHandler` из пакета «Functions».

У данного класса есть метод `getTokens()`, который принимает строковое представление функции и возвращает очередь из токенов<sup>5</sup>, то есть разбивает выражение на токены. Пробелы, переносы строки и табуляции игнорируются.

Также есть метод `removeMinuses()`, принимающий очередь из токенов, которая может быть возвращена вышеописанным методом `getTokens()`, и заменяющий в ней все унарные минусы специальной элементарной функцией «`neg()`». Это сделано для того, чтобы при вычислении значения выражения не возникало неопределенности со знаком «-» (минус), который может быть интерпретирован и как унарная операция, умножающая значение на минус единицу, и как бинарная операция, считающая разность между двумя числами.

В данном приложении поддерживаются константы, такие как  $e^6$ ,  $\pi^7$ ,  $\gamma^8$ ,  $\phi^9$ . За это отвечает метод `substituteConstants()`, который подставляет значения переменных вместо их строкового представления в очереди токенов.

Наконец, существует метод `getPostfix()`, принимающий очередь токенов в инфиксной записи и возвращающий то же выражение, которое он представляет, в постфиксной записи. Преобразование реализовано с помощью алгоритма сортировочной станции. Также, есть его перегруженная версия, принимающая строковое представление функции и пропускающая его через все вышеперечисленные методы.

## 4.2. Математические функции

На рис. 2 представлена архитектура математических функций. Классы из правого столбца описывают функции или вектор-функции от одного или двух аргументов. Они хранят математическое выражение в постфиксной записи, полученное из метода `getPostfix()` класса `ExpressionHandler` и используют его при вычислении значения в определенной точке.

---

<sup>5</sup> Токен (в контексте данной работы) – открывающая или закрывающая скобка, запятая, элементарная функция, число или переменная

<sup>6</sup> [https://en.wikipedia.org/wiki/E\\_\(mathematical\\_constant\)](https://en.wikipedia.org/wiki/E_(mathematical_constant))

<sup>7</sup> <https://en.wikipedia.org/wiki/Pi>

<sup>8</sup> ([https://en.wikipedia.org/wiki/Euler%E2%80%93Mascheroni\\_constant](https://en.wikipedia.org/wiki/Euler%E2%80%93Mascheroni_constant))

<sup>9</sup> [https://en.wikipedia.org/wiki/Golden\\_ratio](https://en.wikipedia.org/wiki/Golden_ratio)

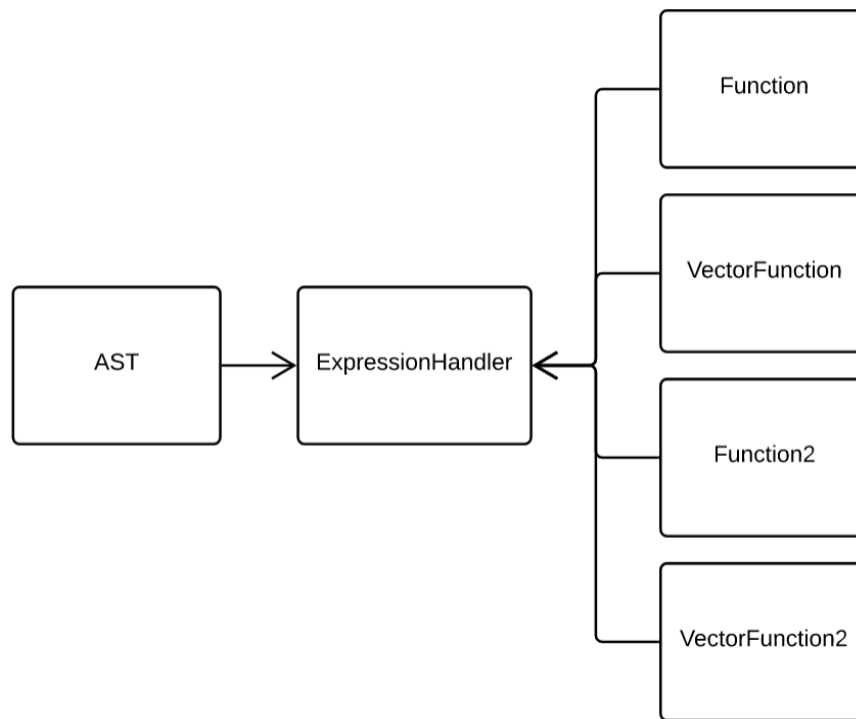


Рис. 2: Архитектура математических функций

Класс AST представляет собой абстрактное синтаксическое дерево. Оно может быть построено по постфиксной записи выражения. В нем реализован метод `reduce()`, который упрощает выражение, которое представляет данное дерево. Под упрощением понимается вычисление выражения в тех местах, где это возможно. Поддерживаются следующие упрощения:

- убирается прибавление, вычитание нуля;
- убирается умножение, деление на единицу;
- умножение на ноль заменяется нулем;
- деление нуля заменяется нулем;
- при возведении единицы в степень показатель убирается;
- возведение в нулевую степень заменяется на единицу;
- возведение в первую степень убирается;
- отрицательное число в чётной степени заменяется на такое же, но с противоположным знаком (становится положительным).

Так, например, выражение « $0*(x-a)+5-3+a*1$ » преобразуется в « $2+a$ ». Это сделано для оптимизации вычислений.

Также, в классе AST есть метод `substitute()`, который подставляет другое полученное выражение вместо указанной переменной в данном выражении.

Наконец, в этом классе реализован метод `diff()`, который перестраивает дерево таким образом, что в итоге получается дерево, представляющее производную прежнего выражения по указанной переменной.

Реализация трёх вышеперечисленных операций без использования дерева – довольно трудная задача, поэтому и было использовано дерево как вспомогательный инструмент.

Для оптимизации классы, представляющие математические функции, хранят выражение также и в сокращенном с помощью класса AST виде. Если в выражении используются много переменных, лишние заменяются нулями. Несокращенное выражение всё же нуждается в хранении, так как у функций поддерживается замена переменных.

Функции поддерживают операции взятия производной и определенного интеграла. У функций от двух аргументов производную можно брать по каждому аргументу, также поддерживается взятие двойного интеграла вместо обычного. У вектор-функций также реализованы операции взятия векторного и скалярного произведения, модуля и нормализации.

Дифференцирование реализовано с помощью класса AST. Интегрирование – через формулу Симпсона. Метод `substitute()` класса AST используется для компактной реализации остальных операций над функциями.

### **4.3. Обработка событий**

Положение камеры задаётся точкой, на которую она направлена, расстоянием до этой точки и специальными углами, называемыми тангаж и рыскание. Непосредственно положение камеры в пространстве высчитывается по формулам в соответствии с этими данными.

Тангаж и рыскание можно изменять, если зажать левую клавишу и начать двигать мышью. Считывается изменение координат мыши по горизонтальной и вертикальной осям и это изменение прибавляется к значениям этих углов. Значение тангажа не может быть больше 90 градусов или меньше -90 градусов.

Расстояние до точки, на которую смотрит камера, изменяется прокруткой колесика. Скорость приближения и удаления можно увеличить, зажав клавишу шифт.

Также, точку, вокруг которой вращается камера, можно сдвинуть, зажав правую кнопку и подвигав мышь. Для удобства эту точку можно сбросить в начало координат, нажав на колесико. Реализовано это следующим образом. По зажатию правой кнопки мыши используются данные о положении камеры для высчитывания векторов, направленных вправо и вверх по отношению к «зрителю». Эти вектора нормируются и откладываются от точки, на которую

смотрит камера. Они служат базисом для задания вектора, на который в конце концов сдвинется камера вместе с точкой, на которую она направлена. Горизонтальная и вертикальная составляющие изменения положения курсора соответственно являются коэффициентами в полученном базисе. Таким образом, первый вектор из базиса умножается на первый коэффициент, второй вектор – на второй коэффициент, после чего полученные вектора складываются. Наконец, точка смещается на полученную сумму этих векторов.

## 4.4. Графический интерфейс

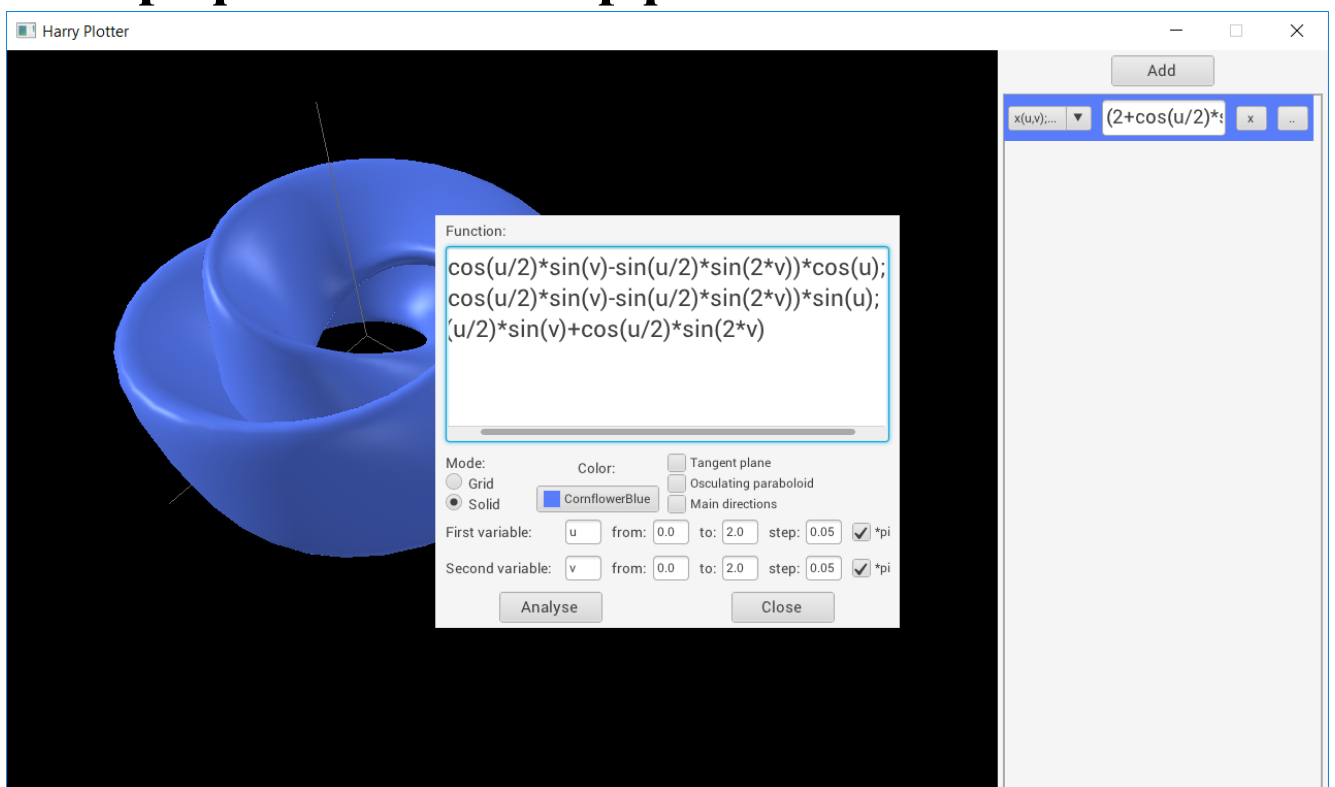


Рис. 3: Графический интерфейс приложения

Графический интерфейс приложения продемонстрирован на рис. 3. В правой части экрана находится список всех добавленных геометрических объектов. Изначально он пуст. Чтобы добавить новый объект, нужно нажать на кнопку «Add». По умолчанию это будет поверхность, заданная явно, но пользователь может указать, что он хочет видеть на экране и его способ задания. Поддерживаются кривая или поверхность, заданные явно или параметрически. В текстовом поле можно вписать уравнение, которым задаётся геометрический объект. Также, можно удалить этот объект и открыть дополнительные настройки.

В дополнительных настройках можно указать цвет, переменные, их области действия, шаг и необходимость умножения параметров на число «пи» для удобства в случае существования тригонометрических формул.

Для кривых указывается дополнительно необходимость отрисовки базиса Френе, а для поверхностей – главных направлений, касательной плоскости и соприкасающегося параболоида, а также способ отображения (как сетку или сплошной). При каждом изменении полей класс Controller считывает новые значения и перестраивает кривую или поверхность.

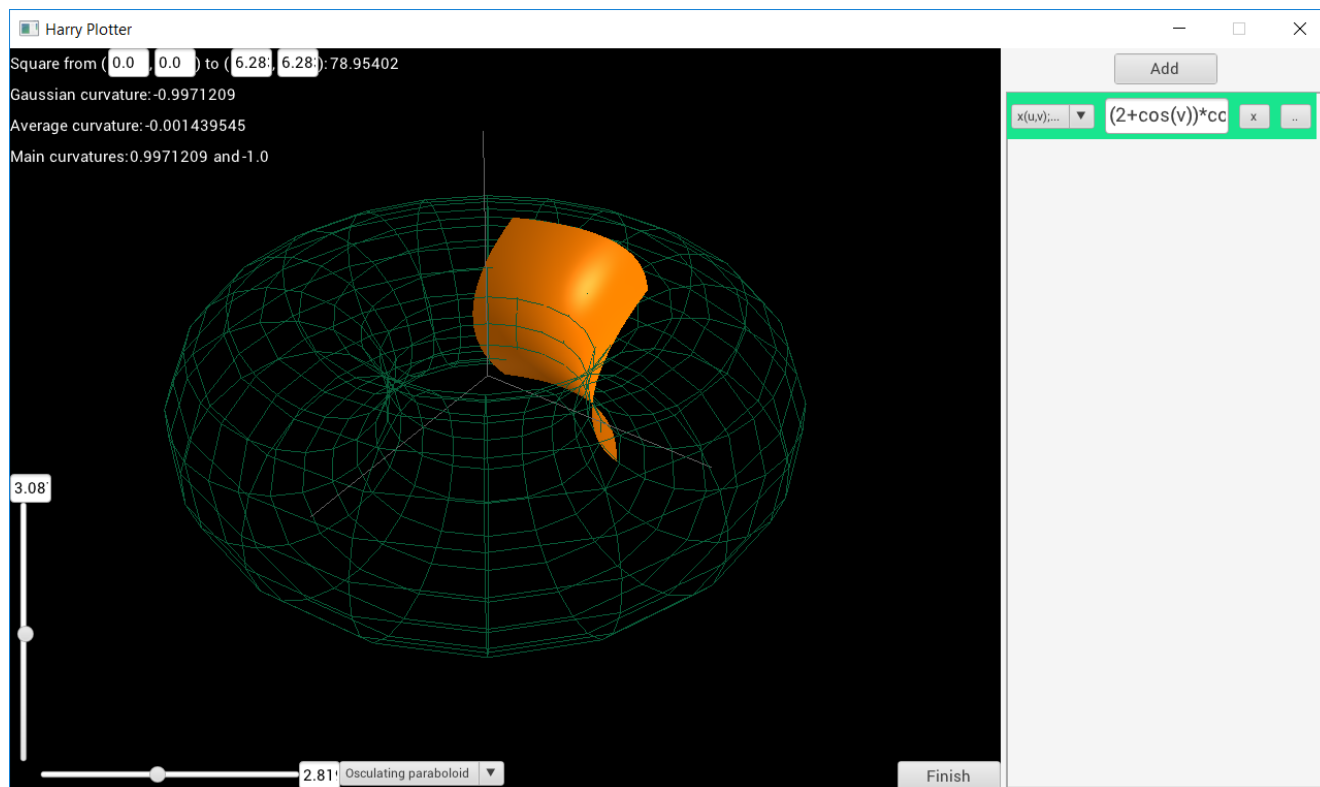


Рис. 4: Графический интерфейс во время анализа поверхности

При нажатии на кнопку «Analyse» Controller вызывает методы у рассматриваемого объекта для подсчёта базовой информации и выводит её на дисплей, а также предоставляет интерфейс для подсчёта длины кривой или площади поверхности. В нижнем углу появляются слайдеры, определяющие точку на кривой или поверхности, которую хочет рассмотреть пользователь. Все другие инструменты считаются относительно текущей выбранной точки. Кнопка «Finish» выводит программу из режима анализа. Интерфейс во время анализа продемонстрирован на рис. 4.

## Заключение

В ходе выполнения данной работы были достигнуты следующие результаты:

- реализован синтаксический анализатор математических выражений;
- реализовано представление математических функций и операции над ними;
- подключена библиотека, предоставляющая графический интерфейс, и создано графическое меню с её использованием.

Таким образом, в результате данного проекта был создан построитель и анализатор кривых и поверхностей с точки зрения дифференциальной геометрии. Код проекта доступен по ссылке<sup>10</sup>.

---

<sup>10</sup> <https://github.com/artemgl/HarryPlotter>

## Список литературы

[1] Цикл видеоуроков по OpenGL (плейлист) – URL:

<https://www.youtube.com/playlist?list=PLRIWtICgwaX0u7Rf9zkZhLoLuZVfUksDP>

[2] Спецификация языка GLSL – URL:

<https://www.khronos.org/registry/OpenGL/specs/gl/GLSLangSpec.4.40.pdf>

[3] Спецификация OpenGL – URL:

<https://www.khronos.org/registry/OpenGL/specs/gl/glspec44.core.pdf>

[4] Библиотека LWJGUI – URL:

<https://github.com/orange451/LWJGUI>