Лицей - Интернат №2

Математическая модель проекции тессеракта и его сечений гиперплоскостью

Авторы:
Ильгиз Мустафин
Гриша Максимов
Артур Нугманов

Научный руководитель:
Марсель Давлетбаев,
преподаватель математики

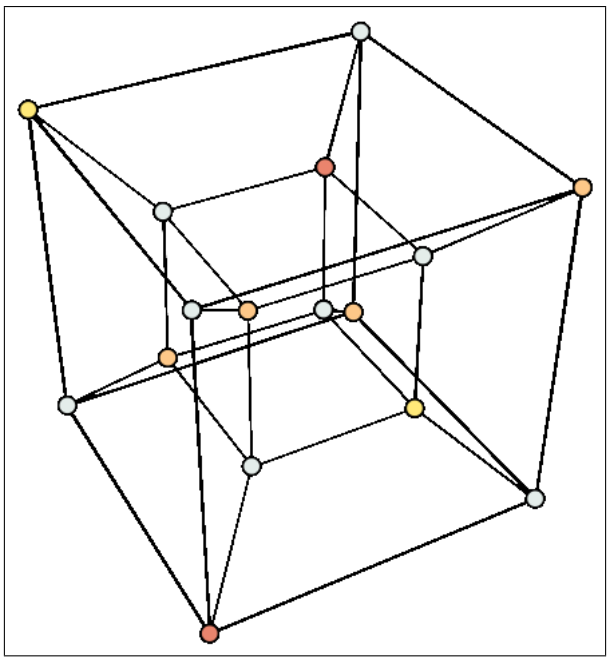
February 6, 2015

Оглавление

1 Введение

Введение, тезисы, примерное описание работы. Наше исследование ставит своей целью разработать программу, которая с помощью введенных данных строит сечение четырехмерного куба - тессеракта трехмерной гиперплоскостью. А также описание работы программы, и ее математическое обоснование. Нами была разработана программа на языке Java, использующая 4 введенные точки в четырехмерном пространстве или вектор и точку для построения трехмерного тела – сечения тессеракта. Для четырехмерного пространства была использована аналогия метода нахождения точек сечения куба плоскостью и метода поворота плоскости в трехмерном пространстве.

Как известно, гиперкуб представляет собой куб в четырехмерном пространстве, может также называться тетракубом и тессерактом. В данной работе будет рассматриваться представление данной фигуры в виде множества точек в Евклидовом пространстве $(\pm 1, \pm 1, \pm 1, \pm 1)$. Тессеракт ограничивают 8 гиперплоскостей, т.е. подпространств, на единицу меньшей, чем объемлющее пространство.



2 Тессеракт, или четырехмерный гиперкуб

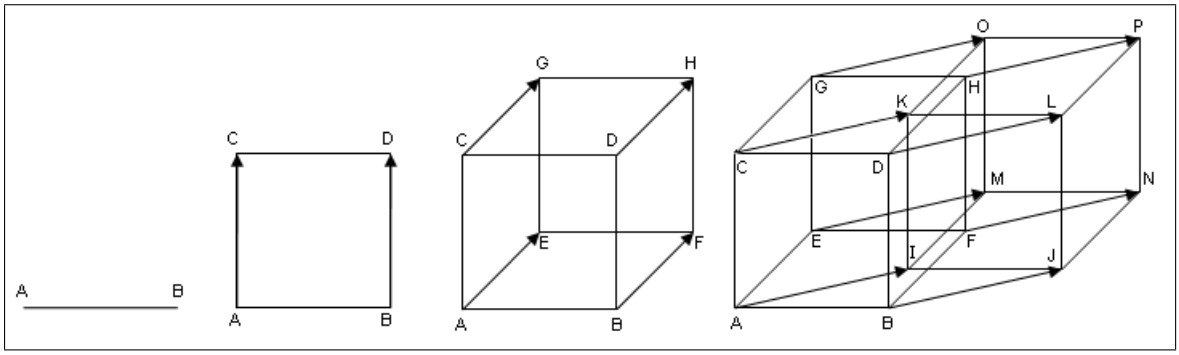
Конечно, невозможно представить графически, как будет выглядеть данный гиперкуб в четырехмерном пространстве, не выходя из трехмерного. Можно лишь представить проекцию данного тессеракта на трехмерную плоскость. В случае проекции невозможно определить точки сечения, поэтому все действия нами выполнялись в четырехмерных координатах для получения трехмерного сечения, представляющего из себя многогранник.

Далее будет рассмотрен довольно простой способ представления проекции тессеракта на трехмерную плоскость. Этот алгоритм позволяет представить примерно, как

выглядит рабочий куб, но, как уже отмечалось выше, он не позволяет определить точки пересечения гиперплоскости с четырехмерным кубом.

2.1 Построение проекции гиперкуба на плоскость

Попытаемся представить себе, как будет выглядеть гиперкуб, не выходя из трёхмерного пространства. Возьмем одномерное пространство (линию), выделим на нем отрезок длиной . Теперь перенесем его на двумерную плоскость и на расстоянии от него нарисуем параллельный ему отрезок той же длины, затем соединим концы. Получится квадрат. Повторив эту операцию с двумерной плоскостью, получим трехмерный куб, а если применить ту же операцию к трехмерному пространству, то получится гиперкуб т.е. его можно представить, как два трехмерных куба, лежащих в параллельных трехмерных пространствах, с попарно соединенными вершинами.



Как видим, отрезок AB в результате параллельного переноса превращается в квадрат, т.е. уже куб 2 измерения. По аналогии куб $ABCDEFGH$ образует проекцию гиперкуба на трехмерную плоскость.

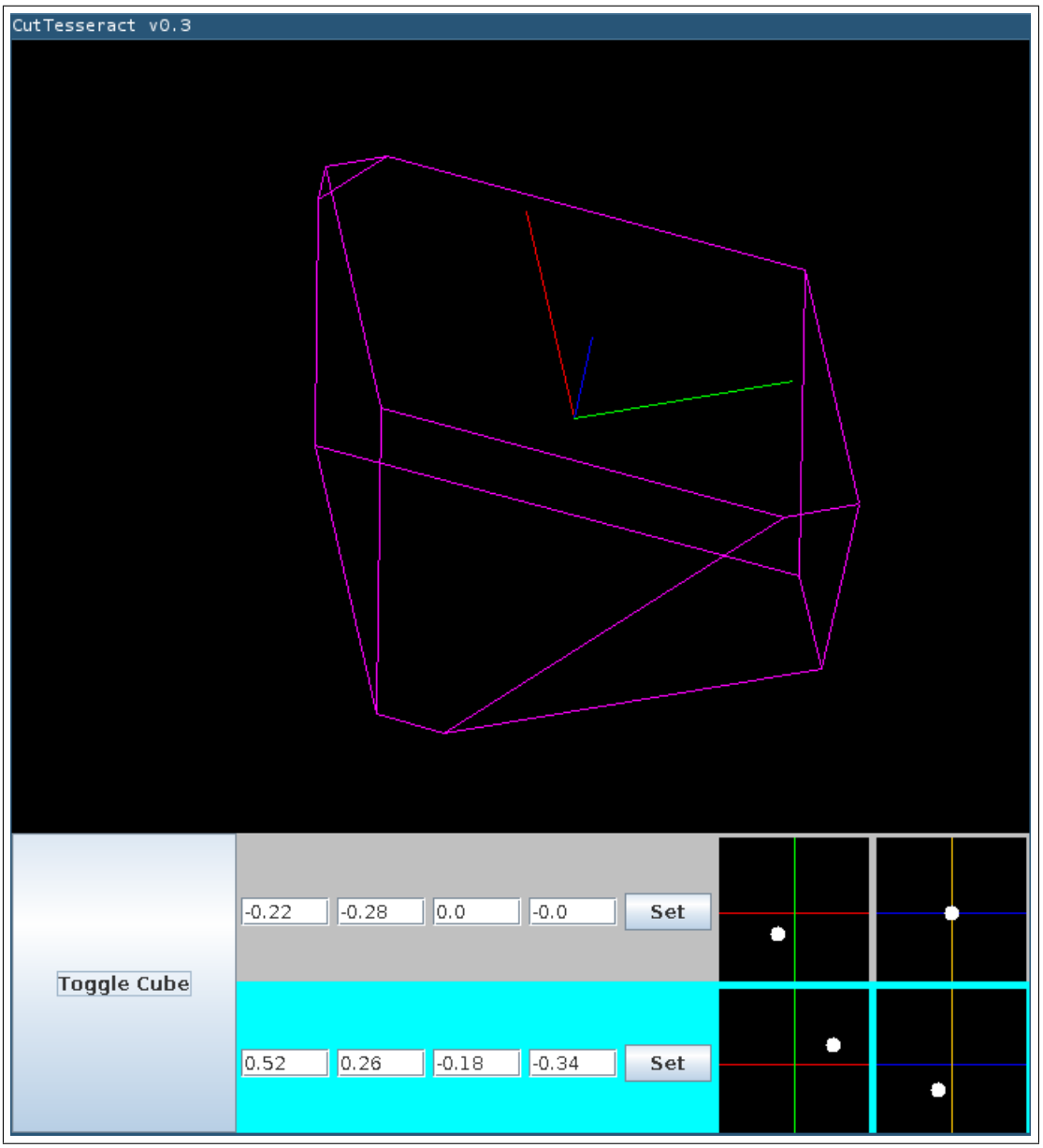
Если задуматься о том, как будет выглядеть сечение тессеракта в четырехмерном пространстве, можно прийти к следующей аналогии: сечением одномерного объекта будет точка, двумерного - отрезок, трехмерного - плоскость. По аналогии сечением четырехмерного объекта является трехмерный объект. Наша задача состоит как раз в отыскании принципа построения данных сечений.

3 Описание работы программы, скриншоты и прочие особенности

3.1 Возможности программы

Данная программа предназначена для визуализации отображения четырехмерного объекта тессеракта на трехмерную плоскость и определения сечения по заданным данным. Для достижения этой цели использовался алгоритм, описанный ниже в пункте "Алгоритм работы программы, его математическое описание".
Как уже было сказано, использовался язык JAVA 7 по причине того, что данный язык программирования представляет возможности очень быстрого прототипирования

и разработки программного продукта, в то же время оставаясь как в высокой степени понятным и дочтаточно легко понимаемым человеком, так и кроссплатформенным. После написания основного алгоритма и логики работы стала задача составить графический интерфейс, для которого была использована графическая библиотека JAVA Swing. Таким образом, перейдем к рассмотрению основных возможностей программы на примере графического интерфейса пользователя:



Главное пространство занимает геометрическая трехмерная модель тессеракта(белые линии) и вписанного в него сечения (на данном скриншоте тессеракт не показан). Ниже панели отображения расположена панель управления, задающая произвольное сечение точкой и вектором нормали. Серая панель задает координаты и местоположение точки относительно основных координатных осей, тогда как голубая панель задает вектор. Как понятно из скриншота, задать точку и вектор, используя графическое окружение можно двумя способами: непосредственно ввести координаты в поля

ввода и нажать кнопку Set, или же использовать для введения специальные квадратные области графического задания координат.

4 Алгоритм работы программы, его математическое обоснование

Перейдем к описанию алгоритма построения сечений.

Для начала нужно задать уравнение гиперплоскости сечения. Для этого нужно либо 4 четырехмерные точки, либо четырехмерный вектор и точку. Таким образом, задается гиперплоскость сечения. После получения уравнения гиперплоскости сечения алгоритм проводит проверку каждого ребра куба на пересечение с гиперплоскостью сечения. Далее проверяем, если точки сечения лежат на одной грани, то можно их соединить (создать из них сегмент), после чего отображаем полученные сегменты на экране по двум точкам и получаем финальное сечение. Однако, оно все еще в четырехмерных координатах, поэтому нормаль гиперплоскости сечения вместе с сечением нужно повернуть так, чтобы она стала параллельна четвертой оси координат (0, 0, 0, 1). В результате этих манипуляций четвертая координата у всех точек сечения становится одинаковой, т.е. её можно не учитывать при выводе полученных граней сечения на экран. После этого сечение представляет собой трехмерное геометрическое тело, вполне доступное для наблюдения и понимания.

Разберем подробнее процесс задания уравнения гиперплоскости сечения.

4.1 Уравнение гиперплоскости сечения

Как было сказано, мы используем два метода для решения данной задачи. Разберем, как задается уравнение гиперплоскости по четырем 4D точкам. Известно, что уравнение искомой плоскости записывается в виде:

$$ax + by + cz + dw + e = 0,$$

где (a, b, c, d, e) - коэффициенты, а (x, y, z, w) - координаты возможных точек. Точка принадлежит гиперплоскости, если выполняется равенство.

Требуется по данным точкам

$$A(x_1, y_1, z_1, w_1), B(x_2, y_2, z_2, w_2), C(x_3, y_3, z_3, w_3), D(x_4, y_4, z_4, w_4)$$

составить уравнение гиперплоскости, т.е. найти нужные коэффициенты.

В программе используется следующее описание четырехмерной точки:

```
public class Point4d implements Comparable<Point4d> {
        /**
        * The <code>x</code> coordinate of this point
        */
        public final double x;
        /**
        * The <code>y</code> coordinate of this point
```

```
*/
public final double y;
/**
* The <code>z</code> coordinate of this point
*/
public final double z;
/**
* The <code>w</code> coordinate of this point
*/
public final double w;
/**
* The point with coordinates <code>(0, 0, 0, 0)</code>
*/
public static final Point4d ZERO = new Point4d(0, 0, 0, 0);
/**
* Constructs point with the specified coordinates
*
* @param x
* the specified <code>x</code> coordinate
* @param y
* the specified <code>y</code> coordinate
* @param z
* the specified <code>z</code> coordinate
* @param w
* the specified <code>w</code> coordinate
*/
public Point4d(double x, double y, double z, double w) {
        this.x = x;
        this.y = y;
        this.z = z;
        this.w = w;
}
/**
* ---FRAGMENT---
*/
```

Как видим, class Point4D содержит одноименный конструктор и 4 координаты x, y, z, w типа double.

Уравнением плоскости будет являться решением данной матрицы:

$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 & w - w_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 & w_2 - w_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 & w_3 - w_1 \\ x_4 - x_1 & y_4 - y_1 & z_4 - z_1 & w_4 - w_1 \end{vmatrix} = 0$$

Далее разберем другой способ. Пусть нам даны точка $A(x_1, y_1, z_1, w_1)$, и вектор

$\vec{N}(x_2, y_2, z_2, w_2)$. С помощью них мы можем задать уравнение гиперплоскости $P(ax + by + cz + dw + e = 0)$.

Программное описание задания точки было дано выше. Сейчас разберем описание четырехмерного вектора:

```
public class Vector4d extends Point4d {
        /**
        * Constructs vector by subtracting begin coordinates from the
        * end coordinates
        *
        * @param begin
        * @param end
        */
        public Vector4d(Point4d begin, Point4d end) {
                super(end.x - begin.x, end.y -
                begin.y, end.z - begin.z, end.w - begin.w);
        }
        public Vector4d(Point4d p) {
                super(p);
        }
        /**
        * Constructs vector with the specified coordinates
        * @param x the specified <code>x</code> coordinate
        * @param y the specified <code>y</code> coordinate
        * @param z the specified <code>z</code> coordinate
        * @param w the specified <code>w</code> coordinate
        */
        public Vector4d(double x, double y, double z, double w) {
                super(x, y, z, w);
        }
        /**
        *---FRAGMENT---
        */
```

Из листинга понятно, что класс вектора есть расширение класса четырехмерной точки, описанной ранее. Данный класс представляет функции конструирования вектора по данным конца и начала или по данным двум точкам.

Вернемся к математическому описанию. Вектор N будет являться нормалью к данной плоскости, т.е. будет справедливо следующее равенство:

$$a = x_2, b = y_2, c = z_2, d = w_2.$$

Останется найти коэффициент e, который будет находиться подстановкой координат данной точки A в уравнение гиперплоскости. Т.е.

$$e = -(x_2x_1 + y_2y_1 + z_2z_1 + w_2w_1)$$

Таким образом, мы нашли гиперплоскость, используя введенные данные.

4.2 Нахождение пересечения гиперплоскости сечения с ребрами гиперкуба

Для нахождения точек сечения надо найти пересечения данной гиперплоскости P и всех ребер куба. Допустим, имеем гиперкуб с координатами

$$(\pm 1, \pm 1, \pm 1, \pm 1),$$

16 вершин или 32 ребра. Для начала проверим, являются ли вершины гиперкуба точками сечения. Для этого берем последовательно все 16 вершин гиперкуба и подставляем их координаты в уравнение гиперплоскости.
$A(x_1, y_1, z_1, w_1)$
$P(ax + by + cz + dw + e = 0)$

$$\left[\begin{array}{ll} ax_1 + by_1 + cz_1 + dw_1 + e = 0 & \text{Вершина гиперкуба принадлежит сечению} \\ ax_1 + by_1 + cz_1 + dw_1 + e > 0 & \text{Вершина гиперкуба лежит выше плоскости сечения} \\ ax_1 + by_1 + cz_1 + dw_1 + e < 0 & \text{Вершина гиперкуба лежит ниже плоскости сечения} \end{array}\right. \quad (1)$$

Теперь, если данные точки, принадлежащие одному ребру, лежат по разные стороны от сечения, то между ними есть точка, принадлежащая сечению. Если обе точки лежат по одну сторону от сечения, то значит, данное ребро и плоскость сечения не имеют общих точек. Возможна также ситуация, когда вершина гиперкуба сама является точкой сечения.

Обратимся к случаю, когда данные точки лежат по разные стороны от сечения. Требуется найти точку пересечения секущей гиперплоскости и данного ребра. Пусть мы работаем с ребром AB гиперкуба и плоскостью сечения $P(ax+by+cz+dw+e = 0)$, для которой мы точно знаем, что она имеет общую точку с ребром.

$A = (x_1, y_1, z_1, w_1)$
$B = (x_2, y_2, z_2, w_2)$
$\vec{AB} = ((x_2 - x_1), (y_2 - y_1), (z_2 - z_1), (w_2 - w_1))(2)$

Зададим параметрическое уравнение прямой AB через точку A и вектор $\vec{AB}$.
$A(x_1, y_1, z_1, w_1)$
$\vec{AB} = (a_x, a_y, a_z, a_w)$

$$\begin{cases} x = x_1 + a_x t \\ y = y_1 + a_y t \\ z = z_1 + a_z t \\ w = w_1 + a_w t \end{cases} \quad (3)$$

Требуется найти неизвестный коэффициент t. Находим его следующим образом, используя уравнение гиперплоскости:

$$P(ax + by + cz + dw + e = 0)$$
$$t = -\frac{(ax_1 + by_1 + cz_1 + dw_1 + e)}{(aa_x + ba_y ca_z + da_w)}$$

Теперь параметрическое уравнение задано, требуется найти координаты точки пересечения данного ребра с полученной прямой. Для нахождения координат точек пересечения нужно вычислить значения (x, y, z, w), входящие в состав системы параметрического уравнения прямой. Найденная точка лежит между вершинами данного ребра тессеракта и принадлежит гиперплоскости сечения.

Требуется повторить данную операцию для всех 32 ребер, которые имеют общие точки с данной плоскостью сечения.

4.3 Построение геометрического тела сечения, поворот

После нахождения всех точек, требуется соединить их ребрами. Для этого перебираем все возможные пары точек из полученного множества. Для каждой пары применим следующие критерии нахождения их на одном ребре - чтобы две данные точки лежали на одном ребре, необходимо, чтобы хотя бы две их координаты были одинаковы и равны соответственно 1 и -1. Если такое условие выполняется, то две данные полученные точки можно соединить ребром. Множество полученных ребер и будет ограничивать все множество точек полученного сечения.

Следующим шагом требуется повернуть плоскость сечения таким образом, чтобы нормаль к данной плоскости стала параллельной четвертой координатной оси Ow. Для осуществления данной цели используются матрицы поворота точки в плоскостях. Составляя нужную композицию поворотов, можно достичь поставленной выше задачи.

Пусть дана система координат $Oxyzw$. Определим матрицы поворота относительно плоскостей:

$$M_{xy}(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} (4)$$

$$M_{yz}(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} (5)$$

$$M_{zw}(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\alpha & -\sin\alpha \\ 0 & 0 & \sin\alpha & \cos\alpha \end{pmatrix} (6)$$

Определим произвольную матрицу поворота $M(\phi)$ как композицию поворотов

$$M(\phi) = M_{xy}(\alpha) \circ M_{yz}(\beta) \circ M_{zw}(\gamma)$$

Таким образом, чтобы повернуть нормаль плоскости сечения параллельно Ow, нам потребуется повернуть последовательно в плоскостях xy, xz, yz.

После данных манипуляций получаем трехмерное результирующее сечение, которое и выводится на экран. Данное сечение не имеет координаты w, и, как следствие, представляет собой геометрическое тело в трехмерном Евклидовом пространстве. Данное сечение имеет одинаковую координату w, и, как следствие, идентично со своей проекцией на трехмерное пространство.

5 Заключение

Таким образом, наше исследование ставило своей целью создание программы, которая может изобразить трехмерное сечение гиперкуба, поворачивать его, используя

формулы поворота точки относительно начала координат, математическое обоснование правильной работы программы и описание её работы. Данная задача была нами достигнута.

Используемые программные средства дали возможность сделать кроссплатформенную версию программы на языке JAVA 7, получить сечения и провести исследовательскую работу над полученными данными.

Результирующий программный продукт может быть запущен как на любой ОС выше Windows XP (включительно), так и на любом компьютере под управлением операционной системы Linux. При этом желательно использование архитектуры linux kernel $\geq$ 2.6.32.

6 Список используемой литературы

https://ru.wikipedia.org/wiki/

7 Приложения

Программный код. Компилировать на Java 7
Программу можно скачать с созданного репозитория по адресу:
http://imustafin.github.io/CutTesseract/