

Исследование сечений тессеракта гиперплоскостью используя методы компьютерного моделирования.

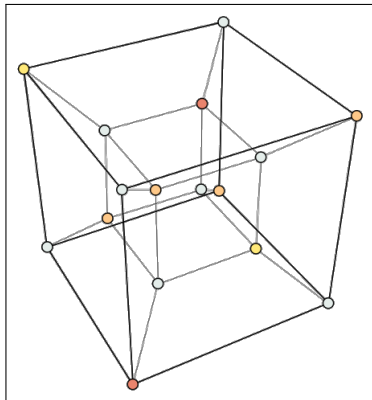
Мустафин Ильгиз, Нугманов Артур, Максимов Григорий

Научный руководитель – Давлетбаев Марсель Фанилевич

МАОУ "Лицей-интернат №2"
Московского района города Казани

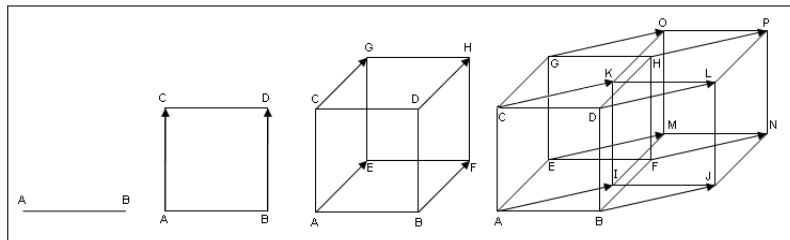
Конференция имени Лобачевского 2015

Тессеракт. Общее определение



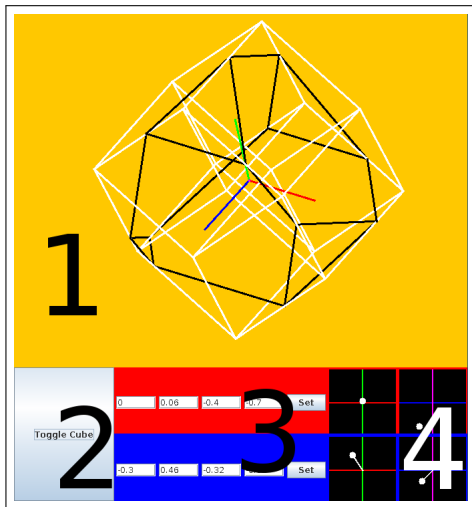
- Тессеракт – куб в четырёхмерном пространстве.
- Рассматриваемая нами модель имеет координаты $(x_1, x_2, x_3, x_4) \in \mathbb{R}^4$, такие, что $x_i \in [-1, 1]$.

Наглядный процесс формирования отображения тессеракта на трехмерную плоскость



Наглядный процесс, как точка *A* переходит постепенно в гиперкуб, приобретая новые размерности

Графический пользовательский интерфейс

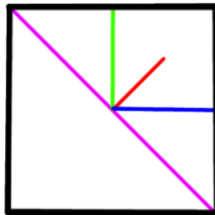


- 1 Панель отображения
- 2 Кнопка переключения отображения тессеракта
- 3 Квадратные области графического ввода координат
- 4 Панели ввода координат

Метод построения сечений тессеракта гиперплоскостью

- Задание гиперплоскость сечения
- Нахождение точки пересечения гиперплоскости и тессеракта
- Поворот получившегося сечения до вложимости его в трехмерное пространство
- Вывод полученного сечения, анализ результатов

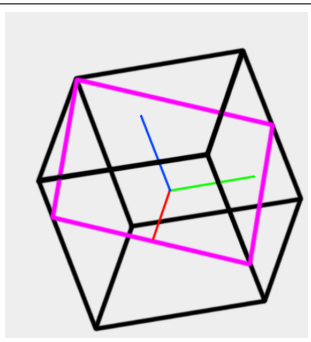
```
Enter input mode
2 for point and normal vector
4 for 4 points4d
2
We need Point4d
0 0 0 0
We need Vector4d as Point4d
1 1 1 1
=====CUT POINTS=====
(-1.0, -1.0, 1.0, 1.0)
(-1.0, 1.0, -1.0, 1.0)
(-1.0, 1.0, 1.0, -1.0)
(1.0, -1.0, -1.0, 1.0)
(1.0, -1.0, 1.0, -1.0)
(1.0, 1.0, -1.0, -1.0)
=====
```



История разработки: CubeCut3d

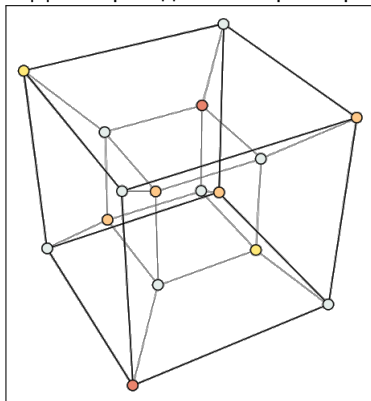
Сначала мы написали программу, которая строит сечение обычного трёхмерного куба двумерной плоскостью. Плоскость задавалась координатами трёх точек, которые задают плоскость сечения. Были реализованы механизмы поворота, масштабирования фигуры. Не использовались никакие сторонние библиотеки для отображения графики, всё было написано с нуля.

```
We need the Cut and the Cube
The Cut is Plane3d. Input 3 Points3d (x1 y1 z1 x
2 y2 z2 x3 y3 z3)...
1 0 0
0 0.5 0
0 0 0.25
Plane3d [a=0.125, b=0.25, c=0.5, d=-0.125]
Have data, will compute
=====CUT POINTS=====
(-1.0, -1.0, 1.0)
(-1.0, 1.0, 0.0)
(1.0, -1.0, 0.5)
(1.0, 1.0, -0.5)
=====
[(-1.0, -1.0, 1.0), (-1.0, 1.0, 0.0)]
[(-1.0, -1.0, 1.0), (1.0, -1.0, 0.5)]
[(-1.0, 1.0, 0.0), (1.0, 1.0, -0.5)]
[(1.0, -1.0, 0.5), (1.0, 1.0, -0.5)]
```



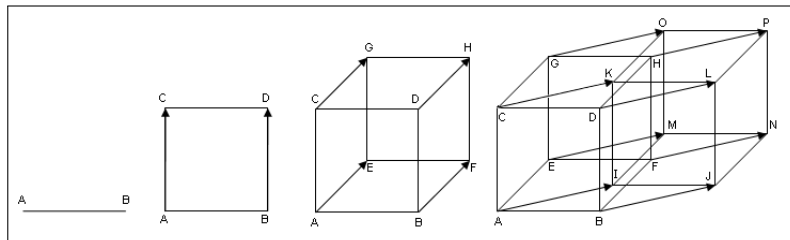
История разработки: DrawTesseract

Для перехода в четырёхмерное пространство нам нужно



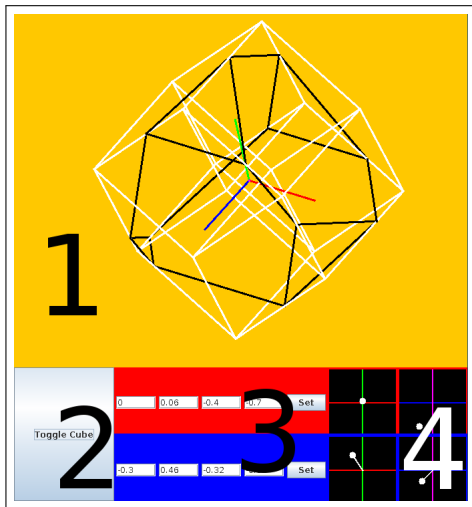
- Рассматриваемая нами модель имеет координаты $(x_1, x_2, x_3, x_4) \in R^4$, такие, что $x_1 \in [-1, 1]$.
- Ограничивается восемью гиперплоскостями
- Имеет 8 трехмерных граней, 24 двумерных, 32 ребра и 16 вершин.

Наглядный процесс формирования отображения тессеракта на трехмерную плоскость



Наглядный процесс, как точка *A* переходит постепенно в гиперкуб, приобретая новые размерности

Графический пользовательский интерфейс

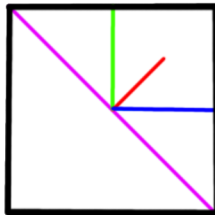


- 1 Панель отображения
- 2 Кнопка переключения отображения тессеракта
- 3 Квадратные области графического ввода координат
- 4 Панели ввода координат

Метод построения сечений тессеракта гиперплоскостью

- Задание гиперплоскость сечения
- Нахождение точки пересечения гиперплоскости и тессеракта
- Поворот получившегося сечения до вложимости его в трехмерное пространство
- Вывод полученного сечения, анализ результатов

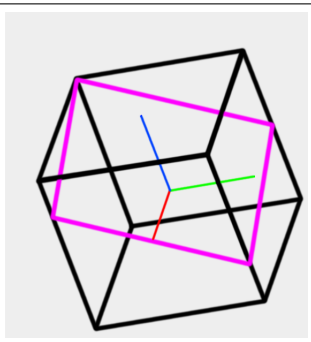
```
Enter input mode
2 for point and normal vector
4 for 4 points4d
2
We need Point4d
0 0 0 0
We need Vector4d as Point4d
1 1 1 1
=====CUT POINTS=====
(-1.0, -1.0, 1.0, 1.0)
(-1.0, 1.0, -1.0, 1.0)
(-1.0, 1.0, 1.0, -1.0)
(1.0, -1.0, -1.0, 1.0)
(1.0, -1.0, 1.0, -1.0)
(1.0, 1.0, -1.0, -1.0)
=====
```



История разработки: CubeCut3d

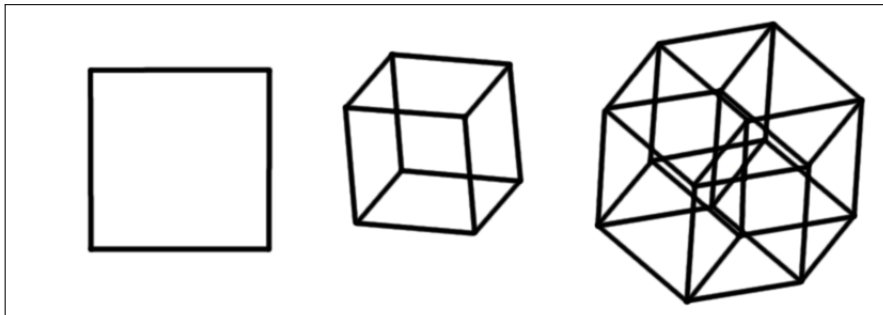
Сначала мы написали программу, которая строит сечение обычного трёхмерного куба двумерной плоскостью. Плоскость задавалась координатами трёх точек, которые задают плоскость сечения. Были реализованы механизмы поворота, масштабирования фигуры. Не использовались никакие сторонние библиотеки для отображения графики, всё было написано с нуля.

```
We need the Cut and the Cube
The Cut is Plane3d. Input 3 Points3d (x1 y1 z1 x
2 y2 z2 x3 y3 z3)...
1 0 0
0 0.5 0
0 0 0.25
Plane3d [a=0.125, b=0.25, c=0.5, d=-0.125]
Have data, will compute
=====CUT POINTS=====
(-1.0, -1.0, 1.0)
(-1.0, 1.0, 0.0)
(1.0, -1.0, 0.5)
(1.0, 1.0, -0.5)
=====
[(-1.0, -1.0, 1.0), (-1.0, 1.0, 0.0)]
[(-1.0, -1.0, 1.0), (1.0, -1.0, 0.5)]
[(-1.0, 1.0, 0.0), (1.0, 1.0, -0.5)]
[(1.0, -1.0, 0.5), (1.0, 1.0, -0.5)]
```



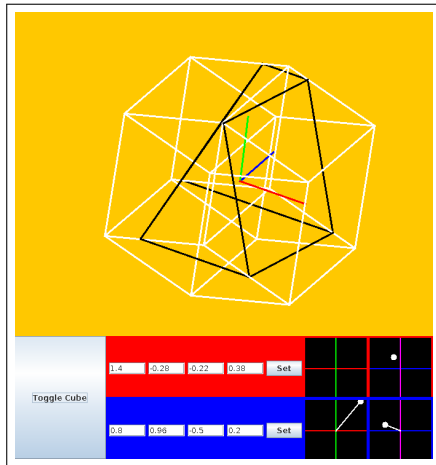
История разработки: DrawTesseract

Для перехода в четырёхмерное пространство нам нужно было научиться строить изображения проекции тессеракта на двумерную плоскость. Так, основываясь на коде первой версии, был написан код программы, которая строит проекцию гиперкуба и позволяет поворачивать изображение как в трёхмерном, так и в четырёхмерном пространстве.

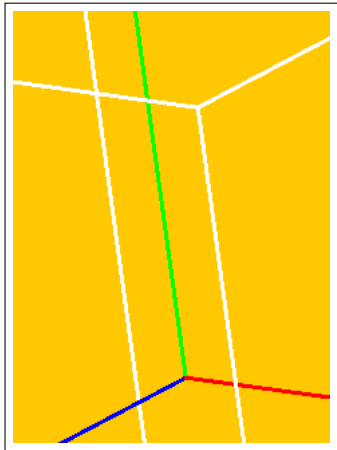


История разработки: CutTesseract

Используя принципы ООП (объектно-ориентированного программирования), было не сложно перейти от рассмотрения трёхмерного к четырёхмерному пространству. Плоскость всё ещё можно было задавать точками, только теперь нужно четыре четырёхмерных точек. Однако, задавать плоскость вектором нормали и точкой на плоскости оказалось удобнее с точки зрения ввода данных. Так и появились Квадратные области графического ввода координат.



История разработки: CutTesseract



В следствие того, что изображение с тонкими линиями было трудно читаемо, нами было принято решение об увеличении толщины линий. Это создало дополнительную задачу для нас: теперь уже был важен порядок рисования линий в связи с тем, что проекции отрезков могут пересекаться и нужно было узнать какой из отрезков ближе к плоскости экрана в точке пересечения трёхмерных отрезков.