МНОГОМЕРНЫЙ АЛГОРИТМ ОВЫПУКЛЕНИЯ РОЯ ТОЧЕК, НАХОДЯЩИХСЯ В НЕОБЩЕМ ПОЛОЖЕНИИ

Выполнил: студент гр. МЕНМ-280901 Корабельников А.А.
Научный руководитель: к.ф.-м.н., доцент Кумков С.С

Институт естественных наук и математики

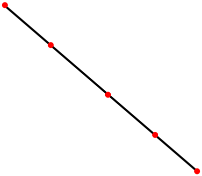
Екатеринбург, 2020

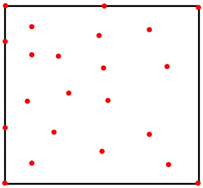
Содержание

Постановка задачи

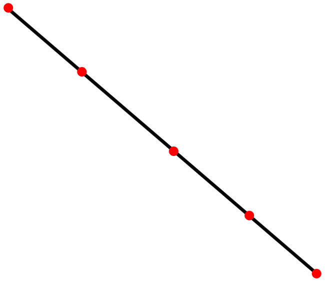
Требуется разработать алгоритм построения выпуклой оболочки многомерного роя точек, находящихся в необщем положении.

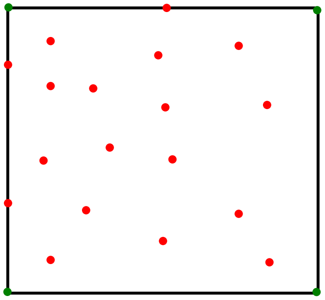
Необщее положение точек означает что в гиперплоскости евклидова пространства размерности n лежит больше чем $n+1$ точка.





Необщее положение точек





Проблемы:

- Требуется вычислять вершины ***(гипер)грани***,
- Требуется вычислять ***(гипер)рёбра*** грани.

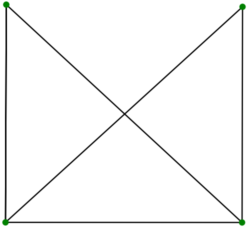
Мне не известны реализации алгоритмов овыпуклениия, работающих в многомерном пространстве в необщем положении.

Алгоритмы построения выпуклой оболочки в nD

Многие алгоритмы для случая плоскости имеют свои аналоги в 3D, но не в большей размерности. Библиотеки вычислительной геометрии:

- CGAL
- LEDA

Основная проблема алгоритмов, нацеленных на общее положение — несимплициальные грани.



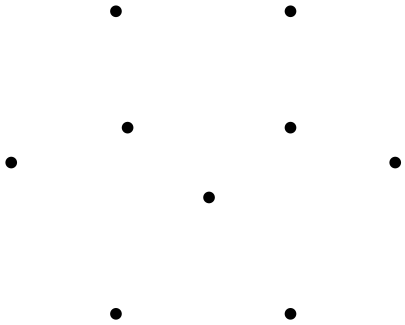
Алгоритмы овыпукления на плоскости

Существует множество алгоритмов овыпукления на плоскости:

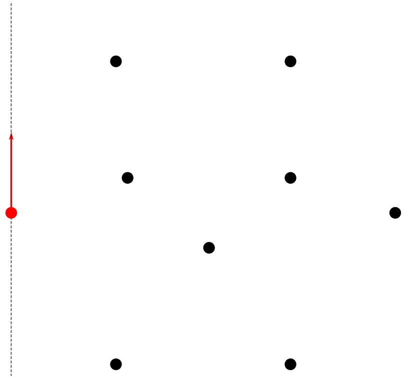
- **Gift wrapping** — $O(nh)$
- Graham scan — $O(n \log n)$
- Quickhull — $O(n \log n)$
- Divide and conquer — $O(n \log n)$
- Monotone chain — $O(n \log n)$
- Chan's algorithm — $O(n \log n)$

Для развития был взят алгоритм заворачивания подарка, т.к. он менее всего использует специфику плоскости.

Построение первой грани

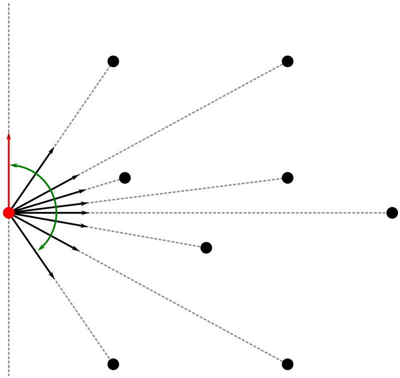


Построение первой грани

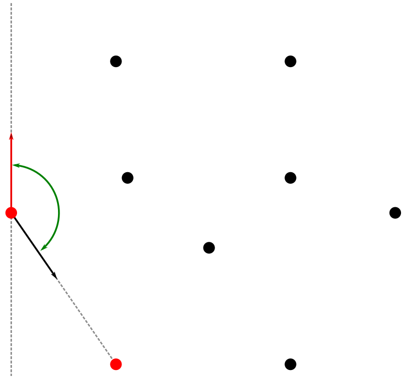


Алгоритм Джарвиса на плоскости

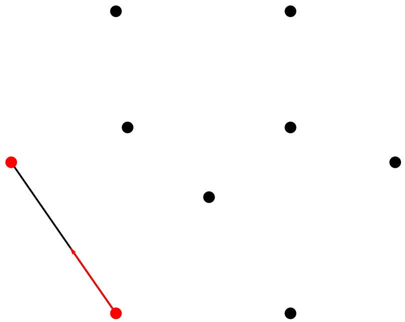
Построение первой грани



Построение первой грани

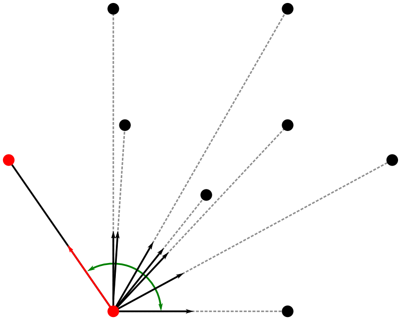


Переход к следующей грани

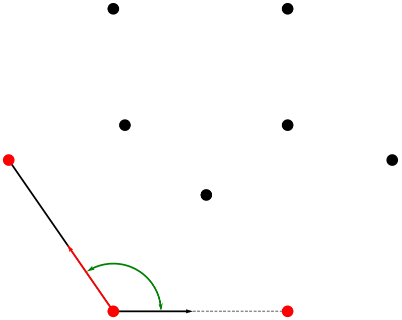


Алгоритм Джарвиса на плоскости

Переход к следующей грани

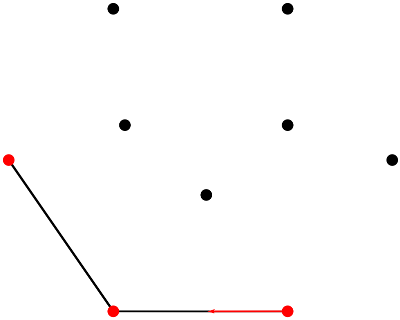


Переход к следующей грани



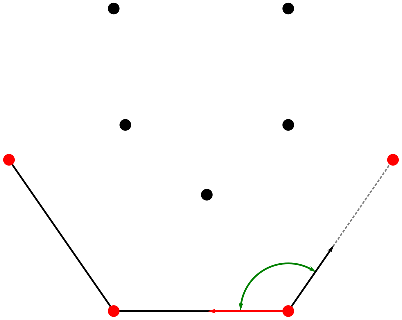
Алгоритм Джарвиса на плоскости

Переход к следующей грани

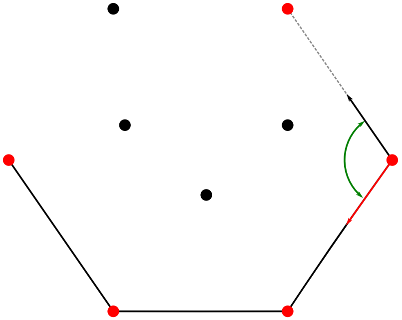


Алгоритм Джарвиса на плоскости

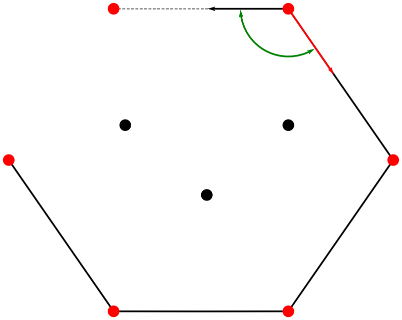
Переход к следующей грани



Переход к следующей грани

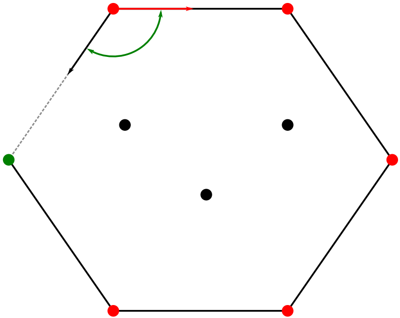


Переход к следующей грани

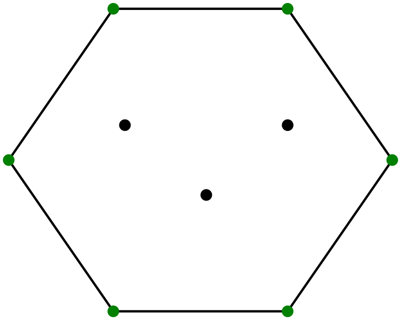


Алгоритм Джарвиса на плоскости

Переход к следующей грани



Конец построения

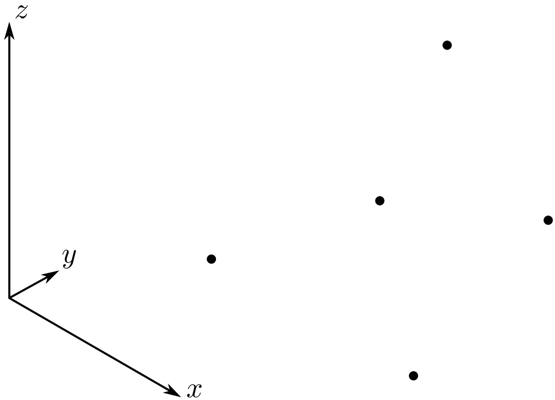


Алгорим Джарвиса при общем положении точек

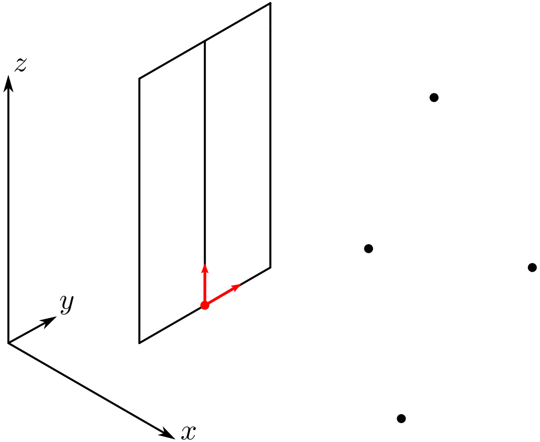
Проблемы расширения:

- поиск первой грани;
- обход граней.

Поиск первой грани

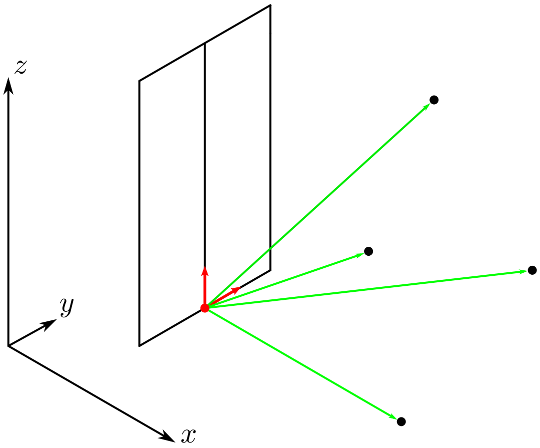


Поиск первой грани

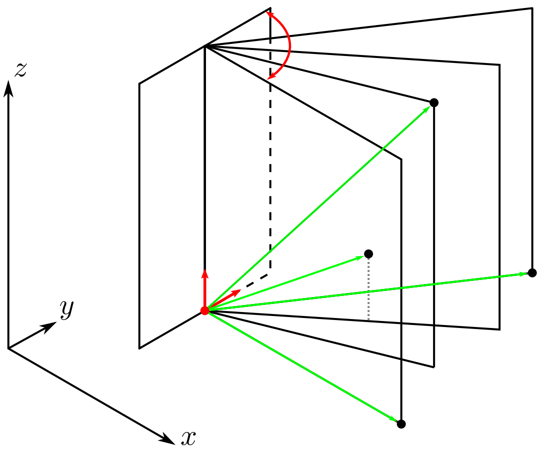


На этом этапе базис плоскости формируется из векторов базиса пространства.

Поиск первой грани

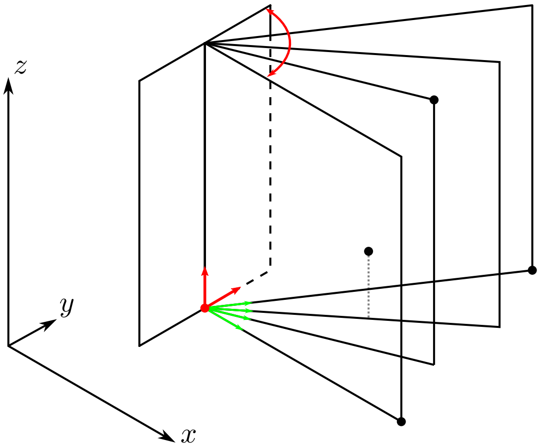


Поиск первой грани

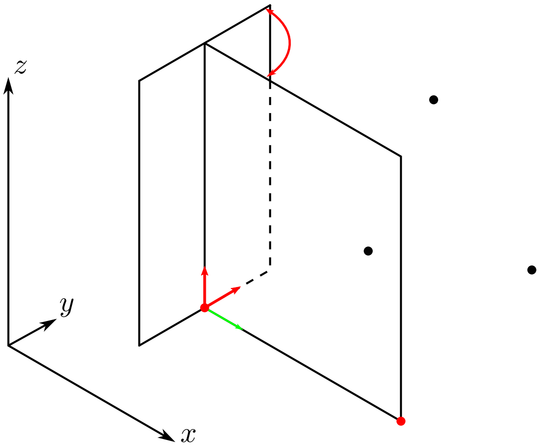


Базис новой плоскости формируется из базиса ребра и вектора в точку.

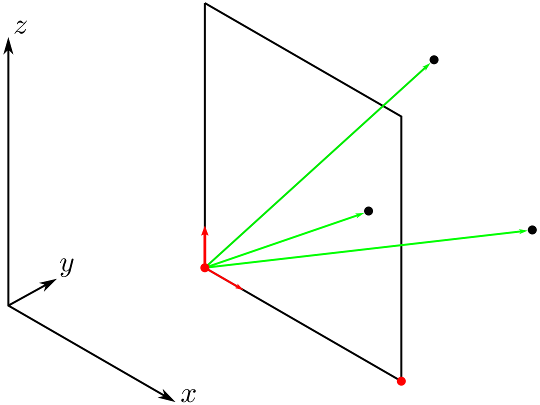
Поиск первой грани



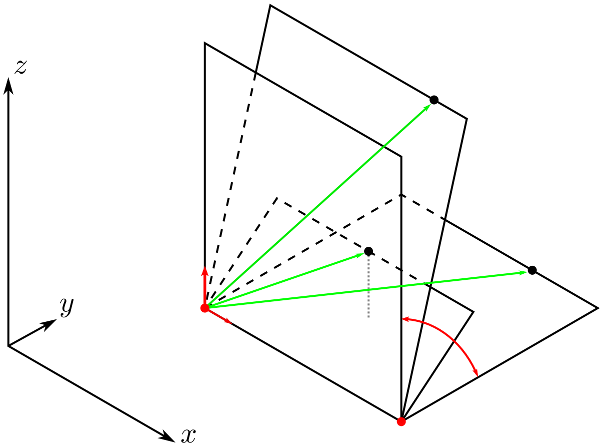
Поиск первой грани



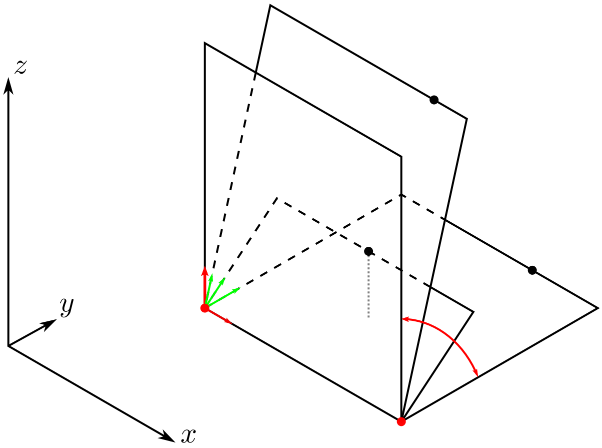
Поиск первой грани



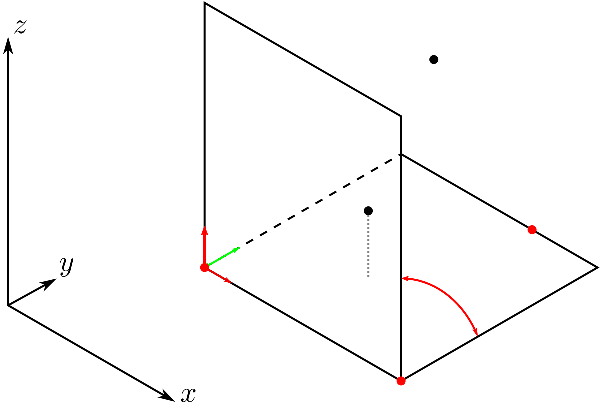
Поиск первой грани



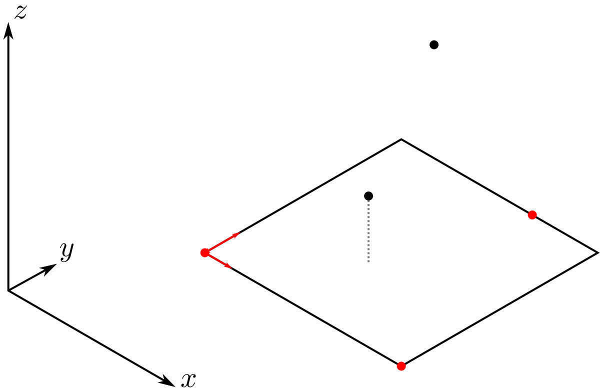
Поиск первой грани



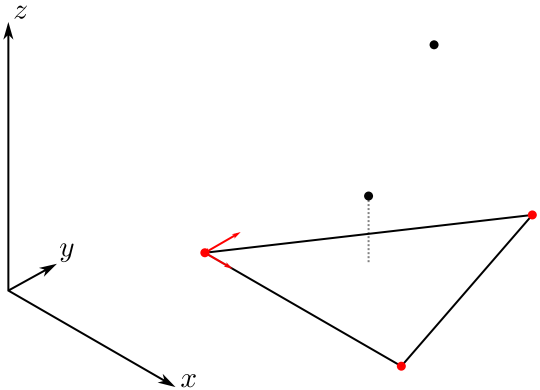
Поиск первой грани



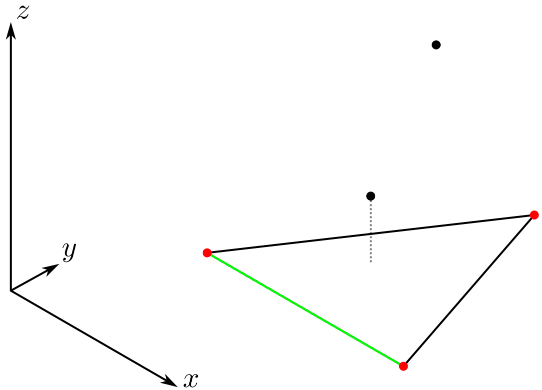
Поиск первой грани



Поиск первой грани

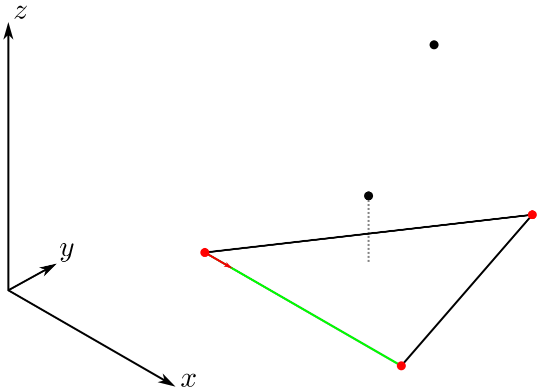


Берем ребро грани

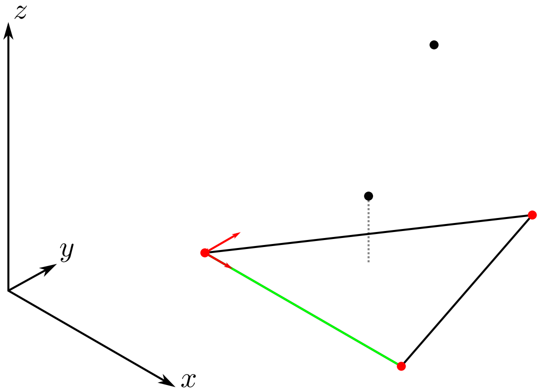


Как происходит переход через ребро

Находим базис ребра

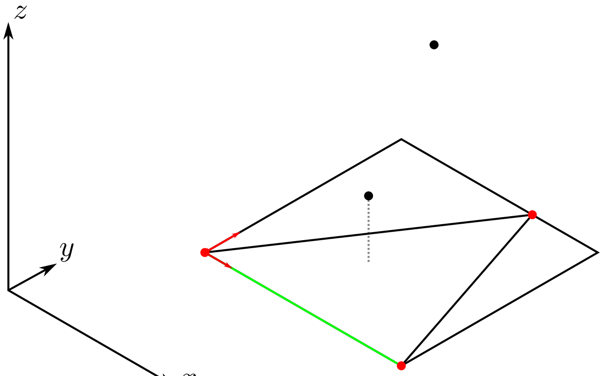


Находим вектор базиса грани



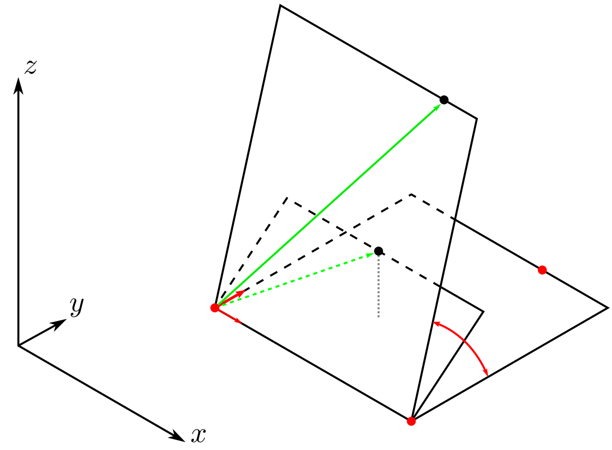
Как происходит переход через ребро

Берем плоскость грани

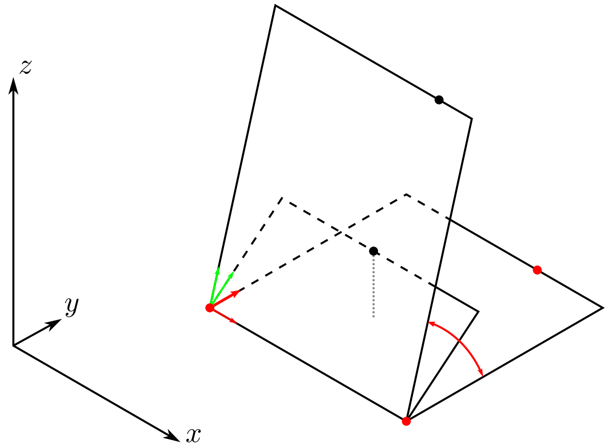


Как происходит переход через ребро

Перебираем свободные точки

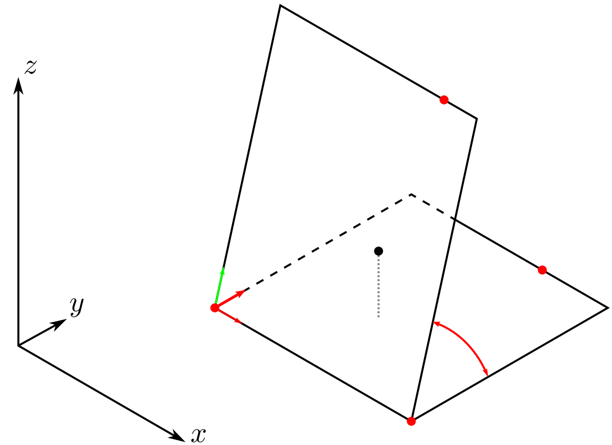


Ортонормируем вектор грани по базису ребра



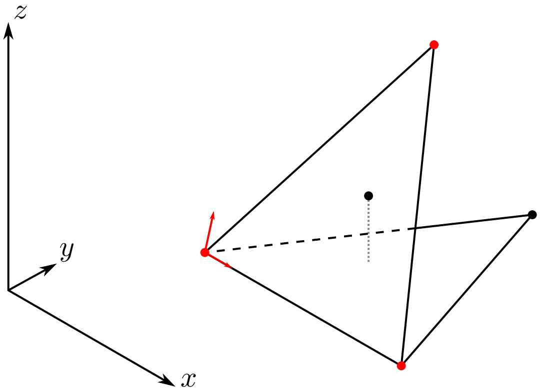
Как происходит переход через ребро

Находим максимальный угол



Как происходит переход через ребро

Переходи выполнен



Обход граней

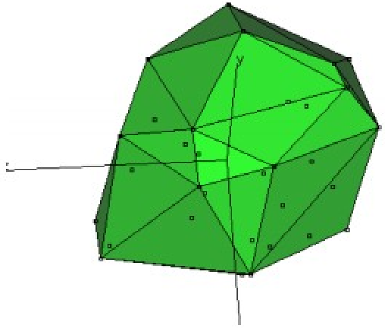
- Перебор ребер - обход в ширину.
- Хранение информации о найденных гранях - хеш-таблица.
- Хеш вычисляется на основе целых чисел, получаемых из коэффициентов уравнения плоскости грани.
  Возможны и другие алгоритмы хэширования граней: на основе точек вершин, на основе индексов точек вершин.

Порядок обхода

- берем необработанную грань;
- для каждого ребра выполняем переход на соседнюю грань;
- если найденная грань еще не обрабатывалась, добавляем в очередь.

Результат

Сложность — $O(n \cdot F \cdot d^2)$,
где F — количество граней, n — количество точек,
d^2 — порядок количества ребер у d-мерного симплекса.



Алгорим Джарвиса при необщем положении точек

Проблемы расширения:

- хранение выпуклой оболочки;
- построение грани;
- обход ребер.

Хранение выпуклой оболочки

Требования к хранению:

- нужно хранить ребра, которые в свою очередь могут быть многомерными несимплициальными многогранниками;
- необходимо хранить список соседних граней;
- нужно хранить информацию о плоскости грани.

Хранение выпуклой оболочки

Требования к хранению:

- нужно хранить ребра, которые в свою очередь могут быть многомерными несимплициальными многогранниками;
- необходимо хранить список соседних граней;
- нужно хранить информацию о плоскости грани.

Грань хранит:

- информацию плоскости: базис, уравнение плоскости;
- список соседних граней;
- структура грани:
  - Если R^d $(d > 2)$, список ребер;
  - Если R^2, список точек;

Проблема построения грани

Проблемы построения:

- априори невозможно указать, какие из точек, попавших в плоскость грани, являются ее вершинами;
- грани выпуклой оболочки могут содержать разное количество ребер.

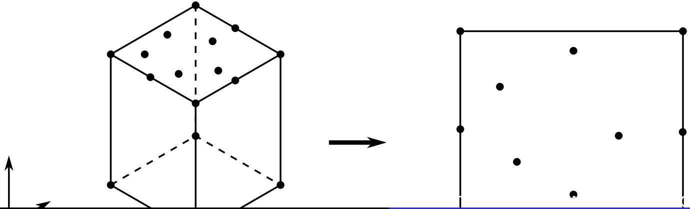
Проблема построения грани

Проблемы построения:

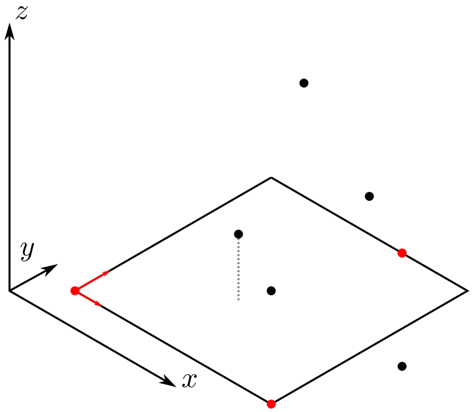
- априори невозможно указать, какие из точек, попавших в плоскость грани, являются ее вершинами;
- грани выпуклой оболочки могут содержать разное количество ребер.

Решение — уход в аффинное подпространство плоскости грани и построение в нем выпуклой оболочки роя точек, попавших в эту плоскость.

Отдельное рассмотрение случая двумерного аффинного подпространства.

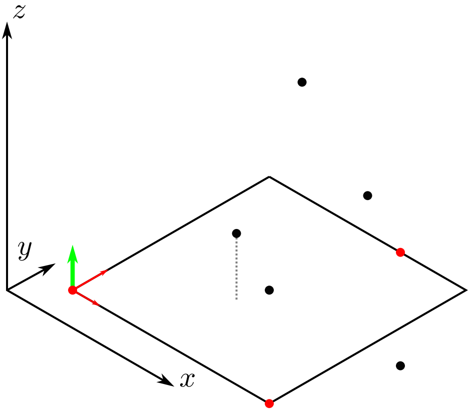


Поиск первой грани



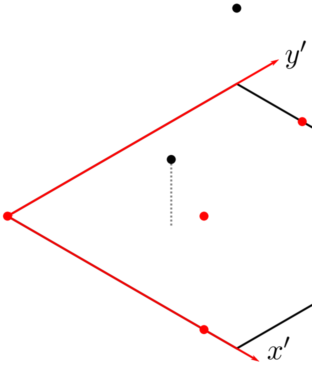
Найти плоскость грани

Поиск первой грани



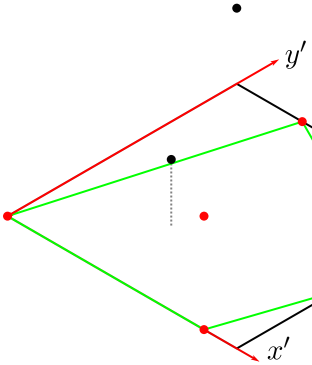
Вычислить нормаль плоскости

Поиск первой грани



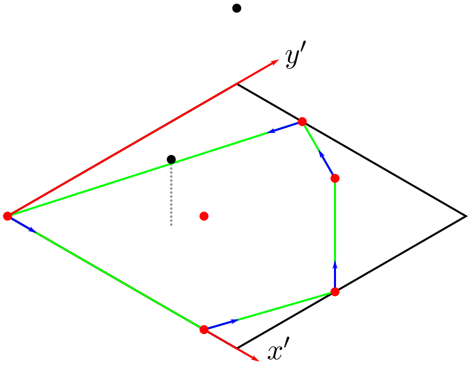
Найти точки и перевести в базис плоскости

Поиск первой грани



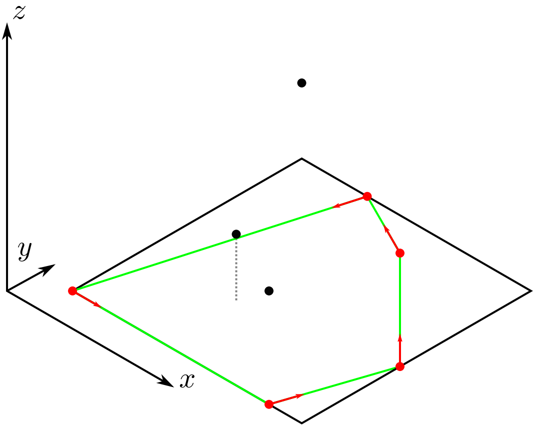
Построить выпуклую оболочку

Поиск первой грани



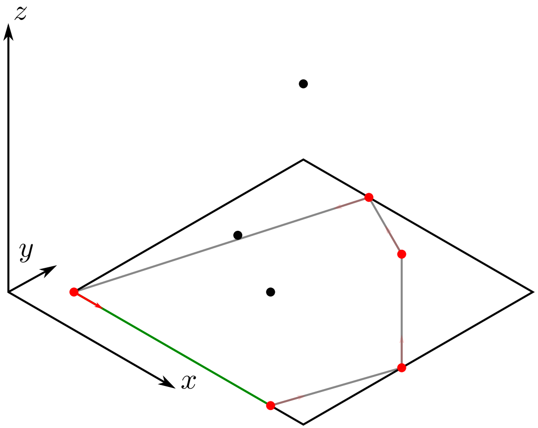
Найти и запомнить базисы ребер

Поиск первой грани



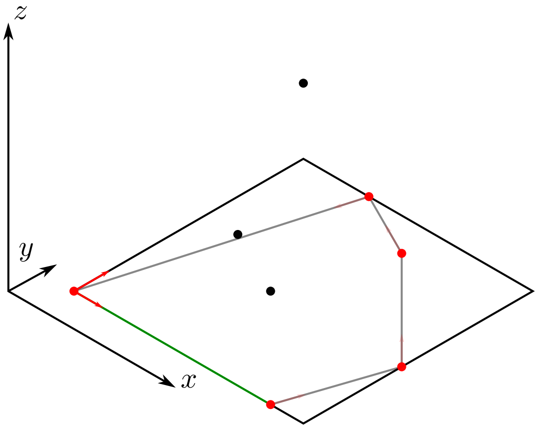
Заменить точки выпуклой оболочки на исходные и пересчитать базисные векторы ребер в координаты исходного пространства. Добавить грань в очередь на обработку.

Переход через ребро



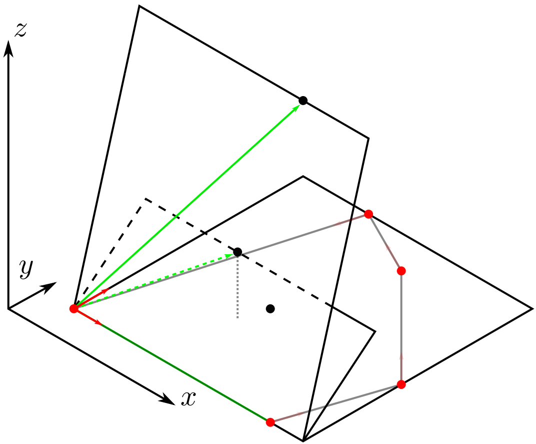
Если очередь граней на обработку пуста, остановить работу.
Иначе взять грань из очереди.

Переход через ребро



Взять очередное ребро обрабатываемой грани.

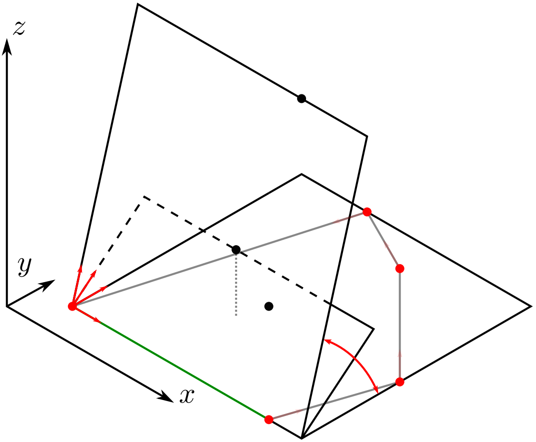
Переход через ребро



Поочередно провести векторы к свободным точкам.

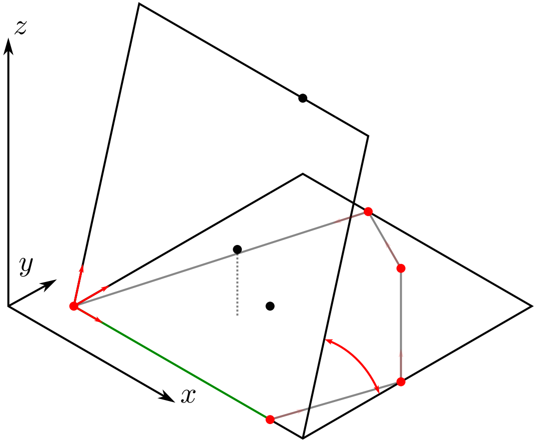
Ортонормировать эти векторы на фоне базиса ребра (шаг процедуры Грама–Шмидта)

Переход через ребро



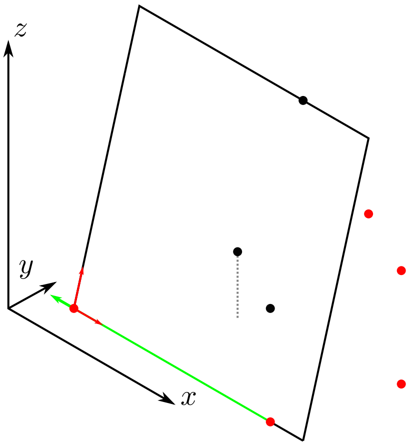
Найти вектор, образующий максимальный угол с вектором грани.

Переход через ребро



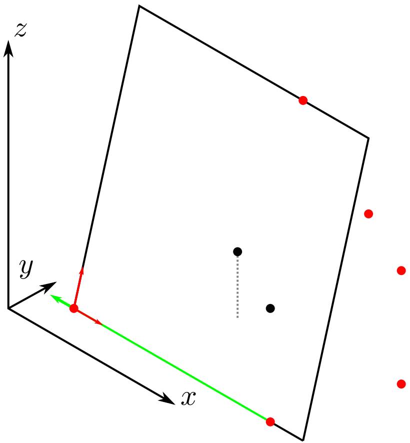
Плоскость грани найдена.

Переход через ребро



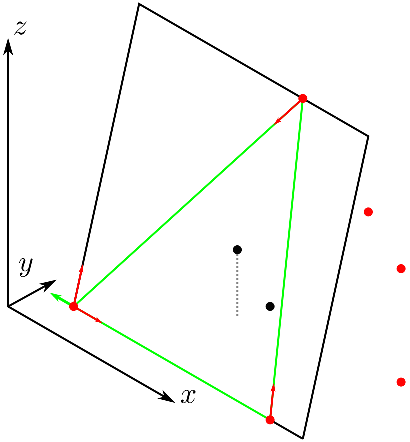
Определить нормаль плоскости. Построить хэш плоскости грани и проверить, обработана ли она уже. Если обработана, запомнить информацию о соседстве граней и перейти к

Переход через ребро



Найти точки, попавшие в плоскость грани. Если их $d+1$ штука, то грань симплициальная и не требует особой обработки. Иначе запустить рекурсивно алгоритм овыпукления

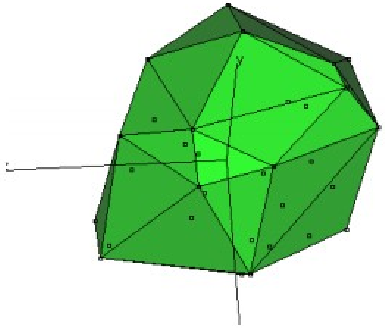
Переход через ребро



Грань построена. Запомнить информацию о соседстве.
Добавить в очередь на обработку.

Результат

Сложность — $O\left((hn)^{d-1}\right)$,
где h — количество ребер, n — количество точек,
d — размерность.



Тестовая реализация

Платформа -.Net Core.
Язык - C#.

Пример работы алгоритма