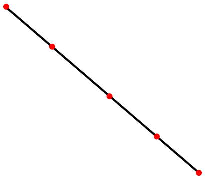
МНОГОМЕРНЫЙ АЛГОРИТМ ОВЫПУКЛЕНИЯ РОЯ ТОЧЕК, НАХОДЯЩИХСЯ В НЕОБЩЕМ ПОЛОЖЕНИИ

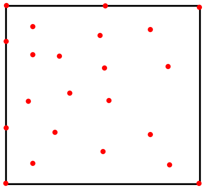
Выполнил: студент гр. МЕНМ-280901 Корабельников А.А.
Научный руководитель: к.ф.-м.н., доцент Кумков С.С

Институт естественных наук и математики

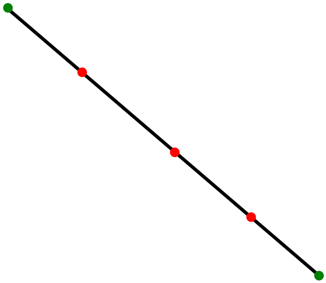
Екатеринбург, 2020

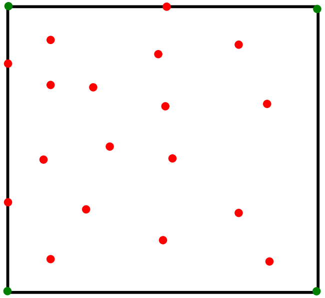
Необщее положение точек означает что в гиперплоскости евклидова пространства размерности d лежит больше чем $d+1$ точка.





Проблема необщего положения





Проблемы:

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

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

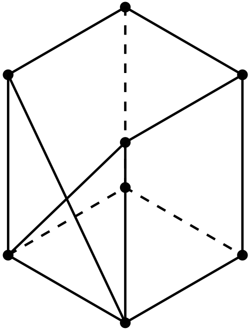
Актуальность

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.



Цель работы

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

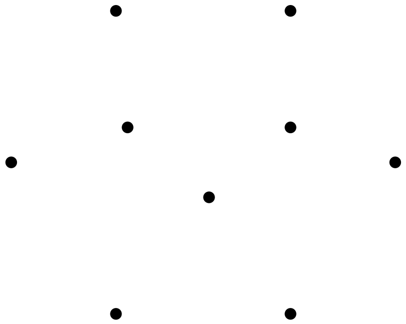
Алгоритмы овыпукления на плоскости

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 — $O(n \log n)$

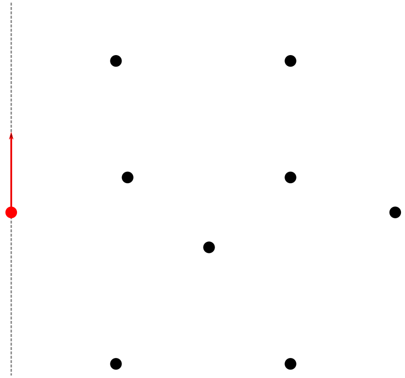
Для развития был взят алгоритм заворачивания подарка, т.к. он менее всего использует специфику плоскости.

Алгоритм Джарвиса на плоскости



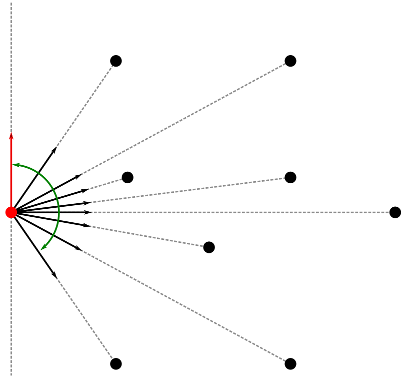
Пусть есть множество точек $P = \{p_1, p_2, \dots p_9\}$, $P \in R^2$.

Алгоритм Джарвиса на плоскости



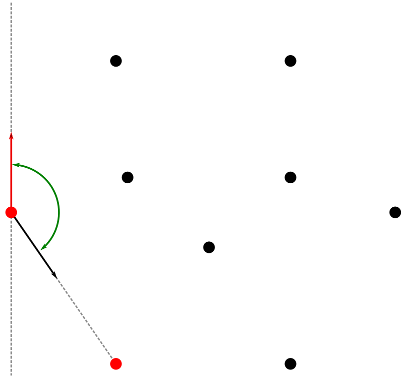
Находим минимальную точку. Проводим через нее прямую, параллельную оси Oy.

Алгоритм Джарвиса на плоскости

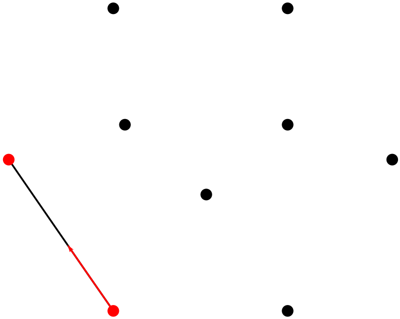


Поочередно проводим векторы к свободным точкам.

Алгоритм Джарвиса на плоскости

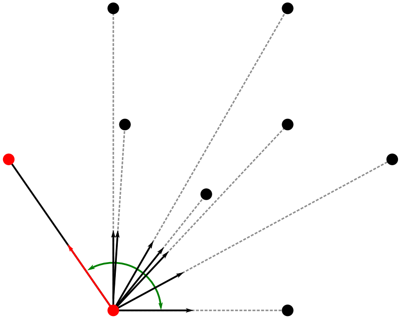


Берем точку, направление на которую образует максимальный угол с направляющим вектором выбранной прямой.



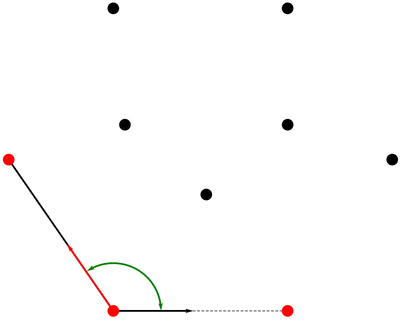
Начальная грань построена. Находим вектор ребра.

Алгоритм Джарвиса на плоскости



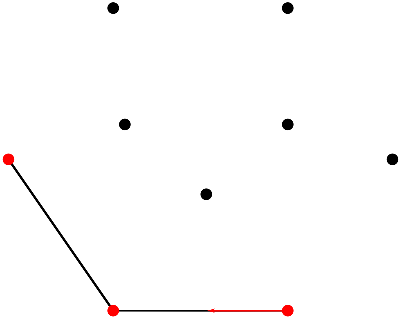
Построение следующего ребра.

Алгоритм Джарвиса на плоскости



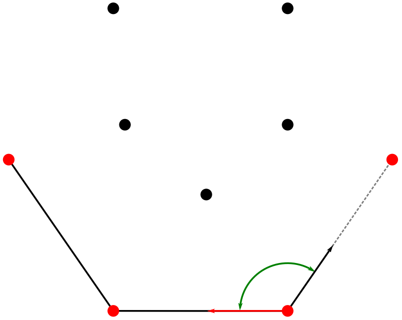
Построение следующего ребра.

Алгоритм Джарвиса на плоскости



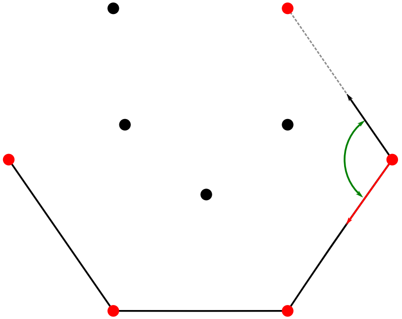
Построение следующего ребра.

Алгоритм Джарвиса на плоскости



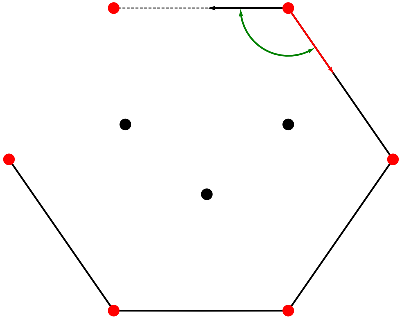
Построение следующего ребра.

Алгоритм Джарвиса на плоскости



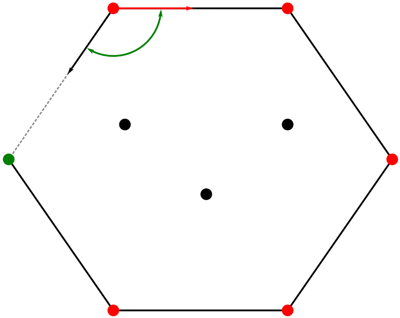
Построение следующего ребра.

Алгоритм Джарвиса на плоскости



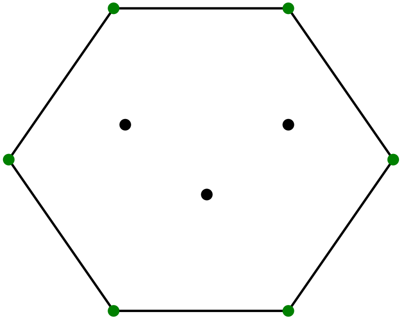
Построение следующего ребра.

Алгоритм Джарвиса на плоскости



Построение следующего ребра.

Алгоритм Джарвиса на плоскости

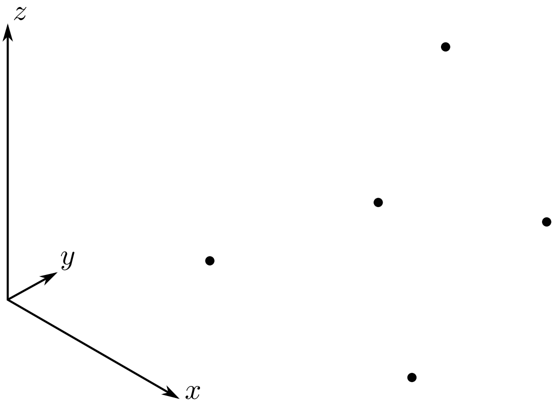


Конец построения

Этапы

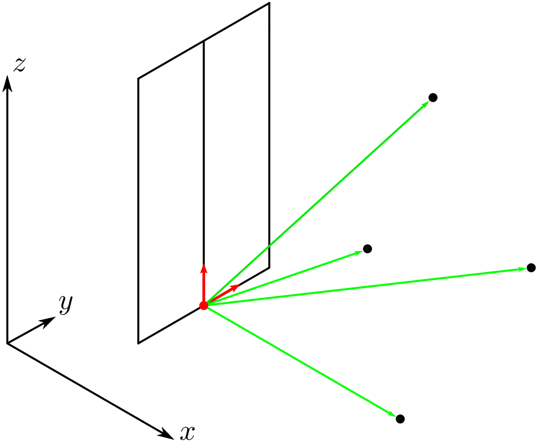
- Разработать реализацию многомерного алгоритма Джарвиса для общего положения точек;
- Расширить реализацию для работы при необщем положеии точек.

Поиск первой грани



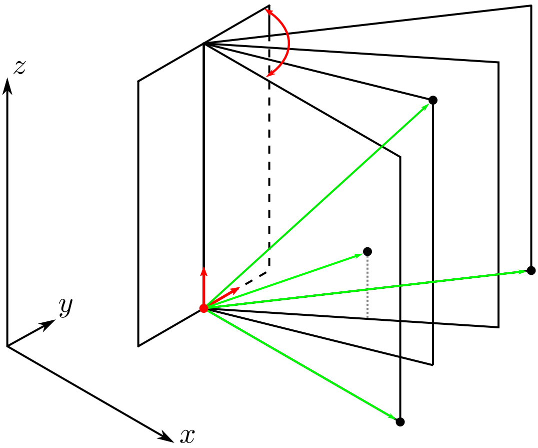
Находим минимальную точку и проводим через нее плоскость, перпендикулярной первой кординатной оси.

Поиск первой грани



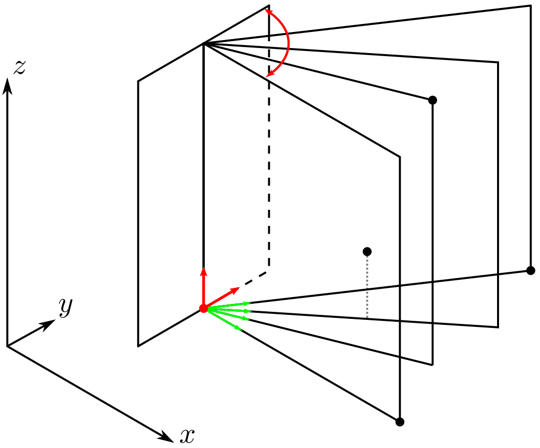
Формируем базис плоскости из векторов базиса пространства. Создаем базис подпространства, исключая первый вектор базиса плоскости.

Поиск первой грани



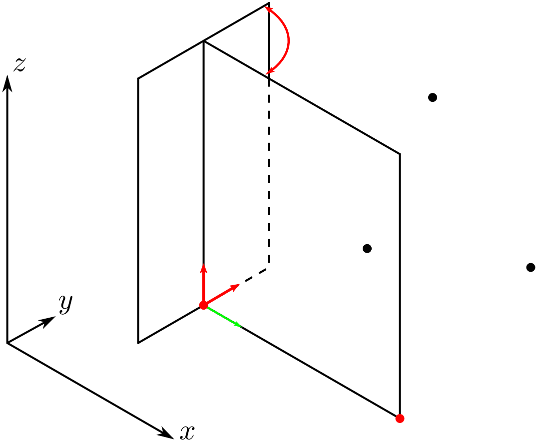
Поочередно проводим векторы к свободным точкам.

Поиск первой грани



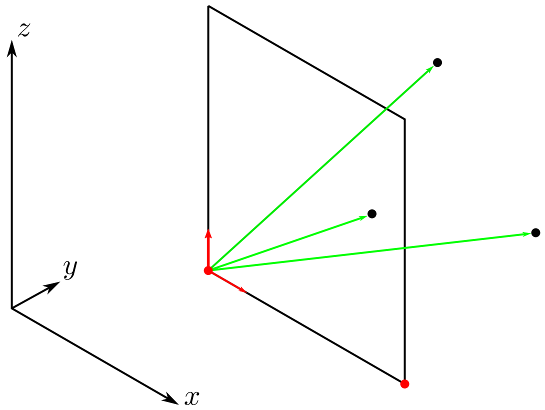
Ортонормируем эти векторы на фоне базиса подпространства (шаг процедуры Грама–Шмидта)

Поиск первой грани



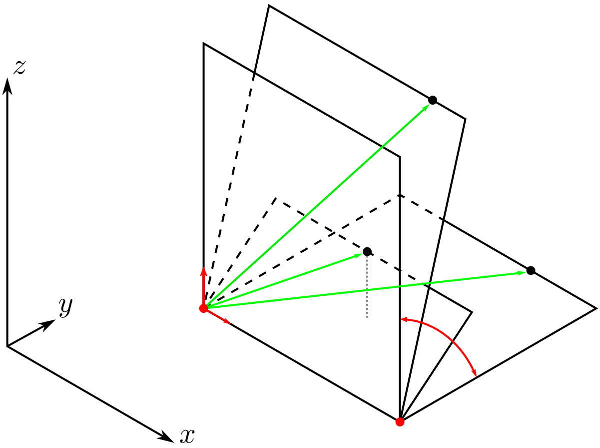
Выбираем вектор, образующий наибольший угол с исключенным вектором базиса. Запоминаем точку. Заменяем исключенный вектор базиса.

Поиск первой грани



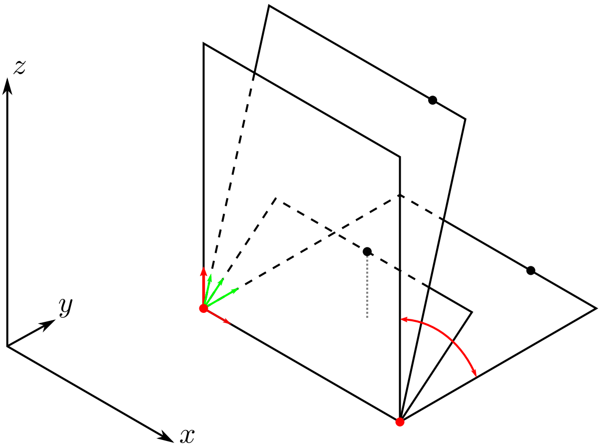
Создаем базис подпространства, исключая второй вектор базиса плоскости.

Поиск первой грани



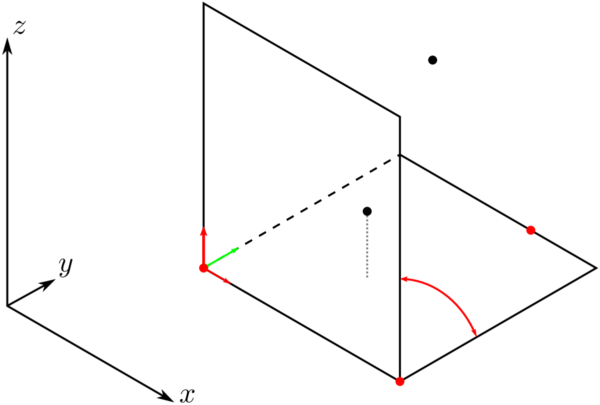
Поочередно проводим векторы к свободным точкам.

Поиск первой грани

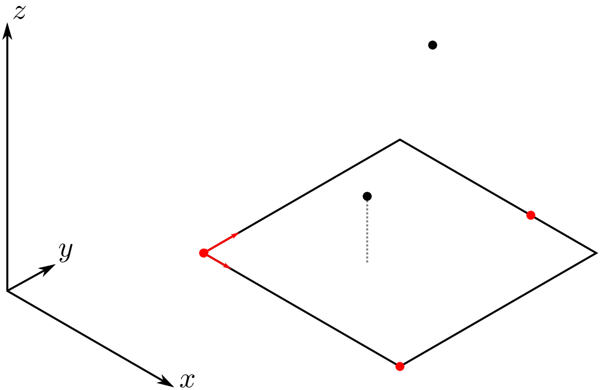


Ортонормируем эти векторы на фоне базиса подпространства
(шаг процедуры Грама–Шмидта)

Поиск первой грани

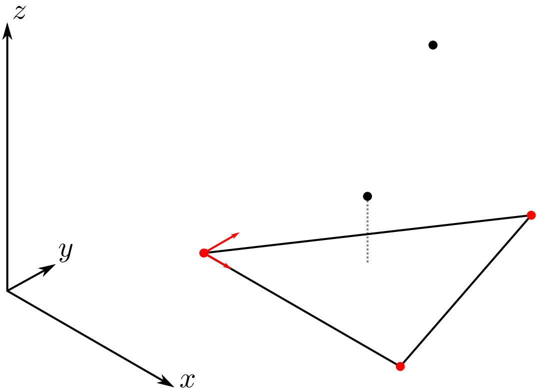


Выбираем вектор, образующий наибольший угол с исключенным вектором базиса. Запоминаем точку. Заменяем исключенный вектор базиса.



После нужного количества повторений этой операции получаем плоскость начальной грани.

Поиск первой грани



Начальная грань найдена. Грань добавляется в очередь на рассмотрение.

Обход граней

- Процесс перебора и создания граней допускает графовую формализацию. Поиск в ширину.
- Хранение информации о найденных гранях — хеш-таблица.
- Хеш вычисляется на основе целых чисел, получаемых из коэффициентов уравнения плоскости грани.

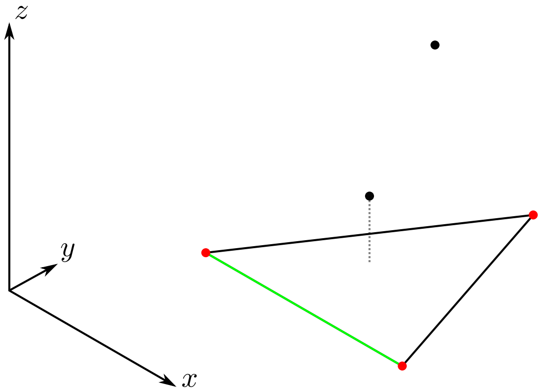
Обход граней

- Процесс перебора и создания граней допускает графовую формализацию. Поиск в ширину.
- Хранение информации о найденных гранях — хеш-таблица.
- Хеш вычисляется на основе целых чисел, получаемых из коэффициентов уравнения плоскости грани.

Порядок обхода

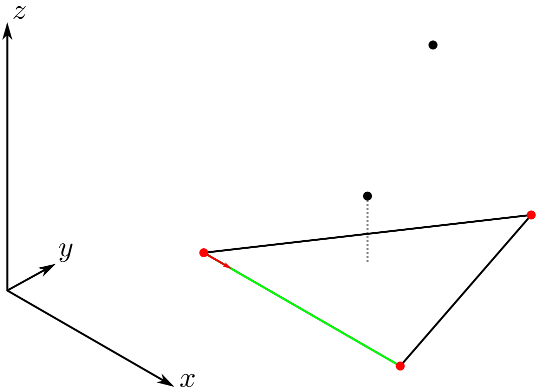
- берем необработанную грань;
- для каждого ребра выполняем переход на соседнюю грань;
- если найденная грань еще не обрабатывалась, добавляем в очередь.

Как происходит переход через ребро



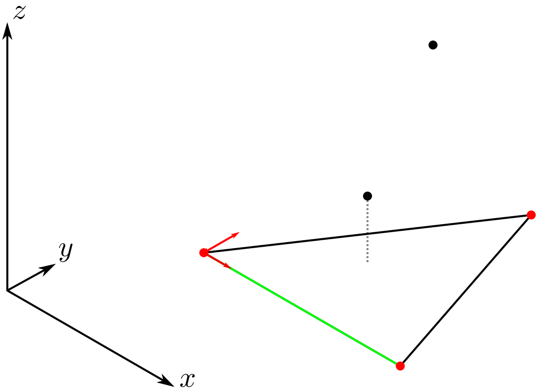
Берем очередное ребро грани и поворачиваем вокруг него плоскость грани.

Как происходит переход через ребро



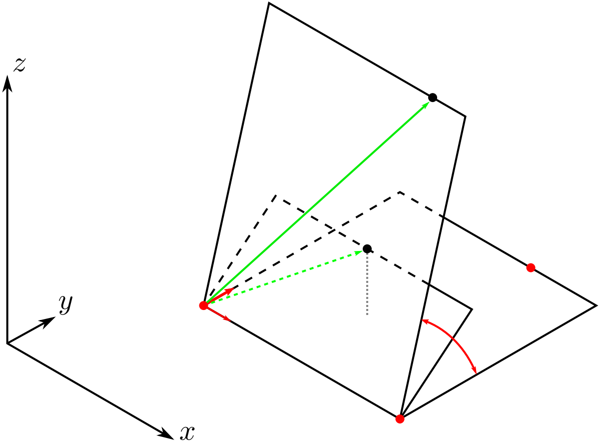
Находим базис ребра.

Как происходит переход через ребро



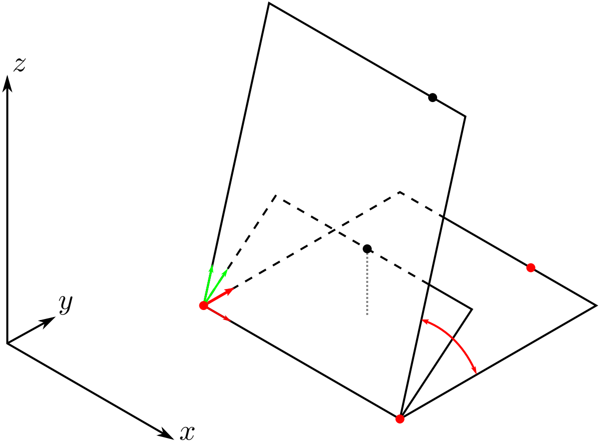
Проводим вектор к свободной точки грани и ортонормируем этот вектор на фоне базиса ребра.
(шаг процедуры Грама–Шмидта)

Как происходит переход через ребро



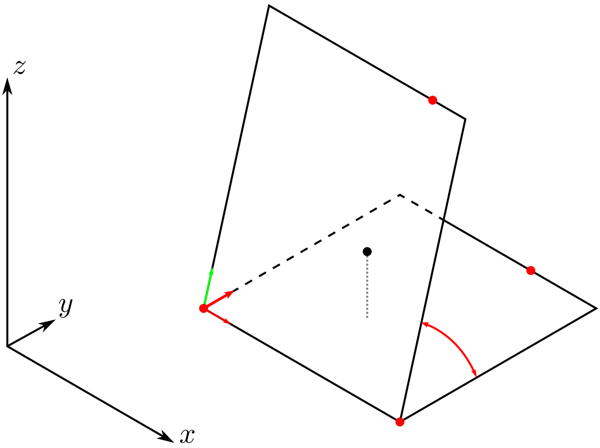
Поочередно проводим вектора к свободным точкам грани.

Как происходит переход через ребро



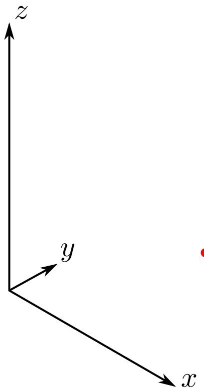
Ортонормируем эти векторы на фоне базиса ребра.

Как происходит переход через ребро



Берем плоскость, образующую максимальный угол с предыдущей гранью.

Как происходит переход через ребро



Грань найдена. Если грань новая, то добавляем ее в очередь на обработку. Переходим к рассмотрению следующего ребра предыдущей грани.

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.

Операция перехода через ребро выполняется d^2 раз для каждой из F граней выпуклой оболочки. При этом перебираются почти все точки роя, которых n штук.

Проблемы расширения на случай необщего положения точек:

- построение грани;

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.

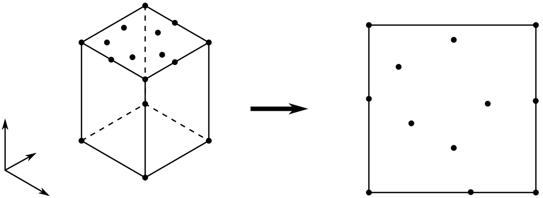
Проблема построения грани

Проблемы построения:

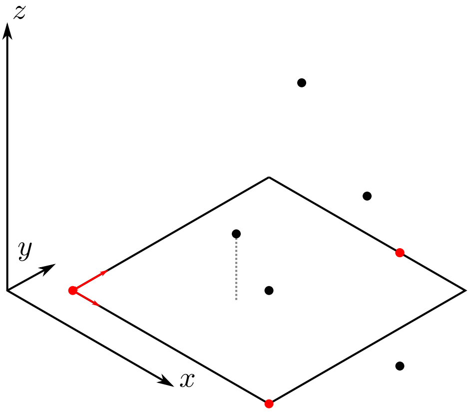
- априори невозможно указать, какие из точек, попавших в плоскость грани, являются ее вершинами;
- грани выпуклой оболочки могут содержать разное количество ребер.

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.

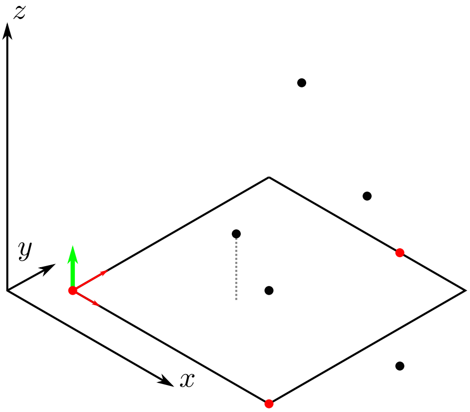


Поиск первой грани



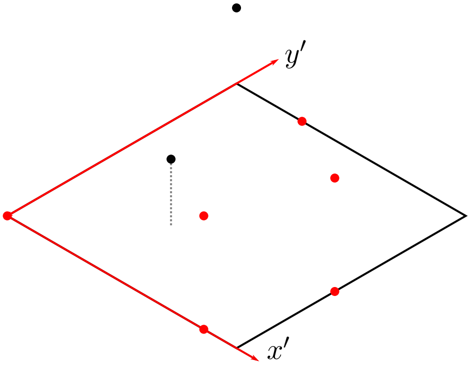
Находими плоскость грани. После этого в сравнении с предыдущим алгоритмом производятся дополнительные шаги.

Поиск первой грани



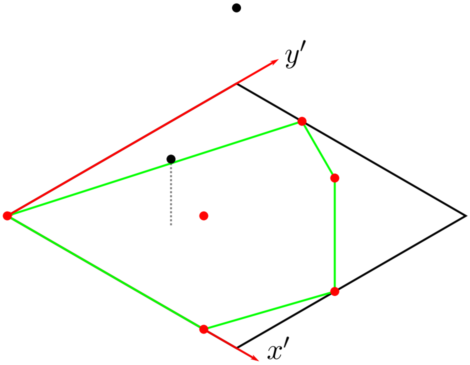
Вычисляем нормаль плоскости

Поиск первой грани



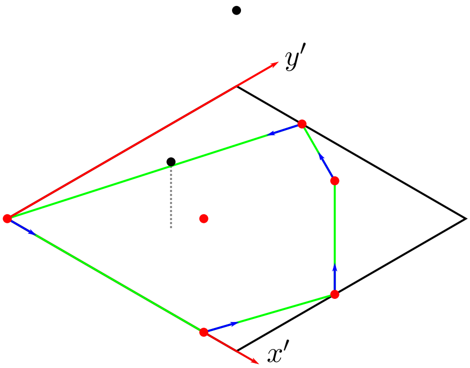
Находим точки и пересчитываем их в базис плоскости

Поиск первой грани



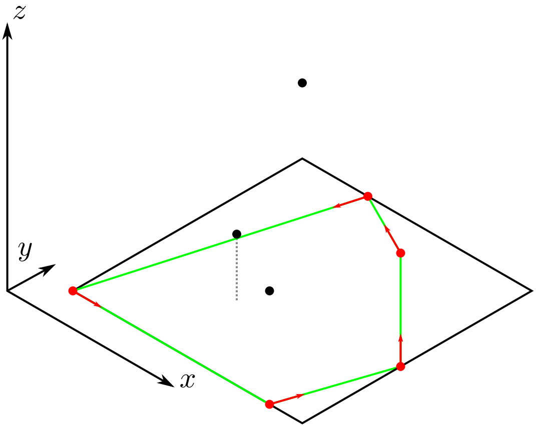
Строим выпуклую оболочку

Поиск первой грани



Если базисы ребер неизвестны, находим. Запоминаем базисы.

Поиск первой грани



Заменяем точки выпуклой оболочки на исходные и пересчитываем базисные векторы ребер в координаты исходного пространства. Добавляем грань в очередь на обработку.

Алгорим Джарвиса при необщем положении точек

Проблемы расширения:

- построение грани;
- хранение выпуклой оболочки;

Хранение выпуклой оболочки

Необходимо хранить:

- Информацию о ребрах: список ребер и их базисы.
- необходимо хранить список соседних граней;
- нужно хранить информацию о плоскости грани:
  - аффинный базис: линейный базис и опорную точку;
  - уравнение плоскости;

Алгорим Джарвиса при необщем положении точек

Проблемы расширения:

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

Переход через ребро осуществляется точно так же, как и в случае общего положения точек. После нахождения плоскости грани, возможен рекурсивный вызов алгоритма.

Сложность полученного алгоритма

Сложность — $O\big((F \cdot n \cdot h)^{d-1}\big)$,
где F — количество граней, n — количество точек, h — количество ребер у грани, d — размерность пространства.

Тестовая реализация

- Платформа -.Net Core.
- Язык - C#.

Результаты тестов

Были проведены замеры времени исполнения:

- 3-мерный тетраэдер с общем положением точек — 0ms (3770 ticks);
- 3-мерный тетраэдер с необщем положением точек (+100 точек)— 5ms;
- 4-мерный симплекс с общем положением точек — 1ms ;
- 4-мерный симплекс с необщем положением точек (+100 точек)–38 ms ;
- 3-мерный куб с лишними точками (+100 точек)— 6ms;
- 4-мерный куб с лишними точками (+100 точек)— 46ms.

В результате работы была разработана реализация алгоритма Джарвиса для многомерного роя точек, находящихся в необщем положении.

В дальнейшем возможна оптимизация как самого алгоритма, так и его реализации.

Спасибо за внимание!

Список литературы

- Preparata F. P., Shamos M. I. Computational Geometry. An Introduction. — New York: Springer, 1985. — 400 с.
- Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. пер. с англ. под ред. Банковского Ю. М. — М.: Мир, 1989. — 478 с.
- de Berg M., Cheong O., van Kreveld M., Overmars M. Computational Geometry. Algorithms and Applications, 3rd Ed. — Berlin: Springer-Verlag, 2008. — 398 с.

Список литературы

- Ивановский С. А., Преображенский А. С., Симончик С. К. Алгоритмы вычислительной геометрии. Выпуклые оболочки в трехмерном пространстве // Компьютерные инструменты в образовани. — 2007. — № 3. — С. 3–17.
- Kokichi S. Robust gift wrapping for the three-dimensional convex hull // Journal of Computer and System Sciences. — 1994. — Vol. 49, issue 2, pp. 391–407.
- Беклемишев Д. В. Курс аналитической геометрии и линейной алгебры. — М.: Наука, 2005. — 308 с.