

1 Обозначения и определения

dD – четырёх и более мерное пространство.

hD – трёх- или более мерное пространство.

Евклидово пространство \mathbb{R}^d , $d > 1$. Его элементы обозначим $\vec{x} = (x_i)_{i=1}^d$.

Ноль-вектор будем обозначать $\vec{0}$.

Аффинное пространство \mathbb{A}^d , $d > 1$. Его элементы обозначим $x = [x_i]_{i=1}^d$.

Элементу $x \in \mathbb{A}^d$ сопоставляется элемент $\vec{x} \in \mathbb{R}^d$, который есть радиус-вектор точки x .

Определены операции умножения на скаляр, сложения и вычитания. Над элементами \mathbb{R}^d они определяются классическим образом как операции линейного пространства. Для аффинного пространства \mathbb{A}^d определения следующие:

- $\forall \alpha \in \mathbb{R} \forall x = [x_i] \in \mathbb{A}^d \alpha \cdot x = \alpha x = [\alpha \cdot x_i] \in \mathbb{A}^d$;
- $\forall a \in \mathbb{A}^d \forall \vec{x} \in \mathbb{R}^d a + \vec{x} = [a_i + x_i] \in \mathbb{A}^d$;
- $\forall a, b \in \mathbb{A}^d a - b = (a_i - b_i) \in \mathbb{R}^d$.

Линейная независимость векторов из \mathbb{R}^d понимается в классическом смысле линейной независимости в линейном пространстве.

Линейная независимость набора точек $\{a_k\}_{k=0}^m \subset \mathbb{A}^d$, $m \leq d$, понимается как линейная независимость набора векторов $\{a_k - a_0\}_{k=1}^m \subset \mathbb{R}^d$.

k -мерным симплексом (k -симплексом) в пространстве \mathbb{A}^d , $k \leq d$, назовём линейно независимый набор $k + 1$ точек.

Базис линейного пространства \mathbb{R}^d понимается в классическом смысле. Аффинный базис аффинного пространства \mathbb{A}^d может представляться в двух эквивалентных формах: либо как d -симплекс этого пространства, либо как пару $(o, \{\vec{x}_i\}_{i=1}^d)$, $o \in \mathbb{A}^d$, $\vec{x}_i \in \mathbb{R}^d$, такую что набор точек $\{o, o + \vec{x}_i\}_{i=1}^d$ линейно независим.

Назовём $(d - 1)$ -грань d -многогранника *симплициальной*, если она имеет ровно d вершин (является $(d - 1)$ -симплексом).

Под словом «плоскость в пространстве \mathbb{R}^d » будем понимать гиперплоскость размерности \mathbb{R}^{d-1} .

Вектор $\vec{e}_i = (\delta_{i,k})_{k=1}^d$ – i -й вектор ортонормированного базиса \mathbb{R}^d . Здесь $\delta_{i,k}$ – символ Крёнекера:

$$\delta_{i,k} = \begin{cases} 1, & i = k, \\ 0, & i \neq k. \end{cases}$$

Обозначим через $\mathcal{N}(\vec{x})$ операцию нормирования вектора: $\mathcal{N}(\vec{x}) = \vec{x} / \|\vec{x}\|$. Считаем, что $\mathcal{N}(\vec{0}) = \vec{0}$.

Обозначим через $\mathcal{ON}(\vec{v}, \mathcal{B})$ операцию ортонормирования вектора \vec{v} на фоне ортонормированного набора векторов \mathcal{B} . Выполняется с использованием алгоритма Грамма-Шмидта. Считаем, что $\mathcal{ON}(\vec{a}, \emptyset) = \mathcal{N}(\vec{a})$.

2 Построение начальной грани \mathcal{P}

Пусть $\mathcal{S} \subset \mathbb{R}^d$ – рой d -мерных точек в d -мерном пространстве. Пусть \mathcal{B} – базис текущей плоскости. Построение начальной грани состоит из двух шагов. Первый – поиск аффинной плоскости, содержащей какую-либо грань выпуклой оболочки \mathcal{P} . Второй – поиск вершин этой грани. Построение аффинной плоскости заключается в последовательном

повороте некоторой начальной плоскости, проходящей через одну точку роя. Каждый поворот аффинной плоскости заключается в подмене одного вектора из её базиса так, чтобы она проходила через ещё хотя бы одну точку роя. Когда очередная плоскость содержит d линейно независимых точек, искомая плоскость построена.

Временные векторы базиса \mathbf{T} – векторы из этого множества могут быть заменены.

Финальные векторы базиса \mathbf{F} – векторы базиса, которые далее не будут заменяться и будут входить в базис искомой плоскости.

Если требуется построить выпуклую оболочку в двумерном пространстве, то запускается какой-либо плоский алгоритм построения выпуклой оболочки. Иначе запускается следующий многомерный алгоритм, вообще говоря, рекурсивный по размерности овыпукляемых роёв.

2.1 Построение начальной плоскости \mathcal{L}

Вход: $\mathcal{S} \subset \mathbb{R}^d$ – рой точек в d -мерном пространстве.

Выход: Начальная плоскость $\mathcal{L} \subset \mathbb{R}^d$, заданная аффинным базисом (o, \mathcal{B}) , где размерность линейного базиса равна $d - 1$. Внешняя нормаль \vec{n} к этой плоскости.

1. Выберем точку $o \in \mathcal{S}$, минимальную в лексикографическом порядке (она гарантировано будет вершиной \mathcal{P});
2. Проведём через $o = (o_i)_{i=1}^d$ плоскость \mathcal{L} перпендикулярно первому базисному вектору пространства. (Все точки роя \mathcal{S} гарантированно лежат не левее этой плоскости.) Положим $\mathbf{F}_0 = \emptyset$, $\mathbf{T}_0 = \{\vec{e}_i\}_{i=2}^d$.
3. Положим нормаль к плоскости $\vec{n}_0 = -\vec{e}_1$.
4. Обозначим \mathcal{V} – множество просмотренных точек плоскости \mathcal{L} . Положим $\mathcal{V} = \{o\}$.
5. Пока $\mathbf{T} \neq \emptyset$, повторяем:
 - (a) Имеем \vec{n}_k – нормаль к текущей плоскости, \mathbf{F}_k – накопленная часть базиса плоскости такая, что (o, \mathbf{F}_k) – аффинный базис тех точек, которые уже просмотрены и включены в плоскость. \mathbf{T}_k – оставшаяся часть временного базиса.
 - (b) Возьмём произвольный $\vec{t} \in \mathbf{T}_k$. Удалим вектор \vec{t} из \mathbf{T}_k : $\mathbf{T}_{k+1} \leftarrow \mathbf{T}_k \setminus \{\vec{t}\}$.
 - (c) Вычислим вектор \vec{v} , перпендикулярный оси вращения: $\vec{v} = \mathcal{ON}(\vec{t}, \mathbf{F}_k)$.
 - (d) Возьмём произвольную точку $s \in \mathcal{S}$, $s \notin \mathcal{V}$. Рассмотрим вектор $\vec{u} = \langle s - o, \vec{v} \rangle \vec{v} + \langle s - o, \vec{n}_k \rangle \vec{n}_k$ – вектор, перпендикулярный оси вращения и лежащий в новой плоскости, содержащей точку s . Если $\vec{u} = \vec{0}$, то есть точка s лежит в оси вращения, то добавляем точку s в множество \mathcal{V} и переходим на шаг 5d.
 - (e) Иначе найдём точку $s_* \in \mathcal{S}$ такую, что $s_* \notin \mathcal{V}$, и угол между \vec{v} и \vec{u} , наибольший среди всех точек роя (то есть скалярное произведение $\langle \vec{v}, \mathcal{N}(\vec{u}) \rangle$, наименьшее среди всех таких точек из \mathcal{S}).
 - (f) Если точка s_* не нашлась, это означает, что весь рой \mathcal{S} лежит в аффинном подпространстве размерности меньше $d - 1$. В этом случае или алгоритм прекращает работу, если целью было найти выпуклую оболочку полной размерности, или переходит к построению выпуклой оболочки роя \mathcal{S} в найденном аффинном подпространстве с базисом (o, \mathbf{F}_k) .
 - (g) Если таких экстремальных точек несколько, то можно выбрать любую. Расширим финальный базис: $\mathbf{F}_{k+1} \leftarrow \mathbf{F}_k \cup \{\mathcal{ON}(s_* - o, \mathbf{F}_k)\}$. Добавим точку s_* в множество \mathcal{V} .

- (h) Вычислим нормаль \vec{n}_{k+1} новой плоскости: $\vec{n}_{k+1} = \langle \vec{u}^*, \vec{n}_k \rangle \vec{v} - \langle \vec{u}^*, \vec{v} \rangle \vec{n}_k$. При необходимости надо переориентировать \vec{n}_{k+1} так, чтобы $\langle \vec{n}_{k+1}, s - o \rangle < 0$, где s — точка роя, не лежащая в текущей плоскости, то есть такая, что $\langle \vec{n}_{k+1}, s - o \rangle \neq 0$. Если такой точки не нашлось, значит все точки лежат в аффинном подпространстве с базисом (o, \mathbf{F}_{k+1}) . Аналогично пункту 5f либо останавливаем алгоритм, либо переходим к построению выпуклой оболочки в этом аффинном подпространстве.

6. Положим $\mathcal{B} = F_k$, $\vec{n} = \vec{n}_k$.

2.2 Построение грани

Вход: Рой точек $\mathcal{S} \subset \mathbb{R}^d$ и плоскость $\mathcal{L} \subset \mathbb{R}^d$, то есть аффинный базис (o, \mathcal{B}) . Возможна передача $(d - 2)$ -ребра \mathcal{E} , являющегося начальной гранью для процедуры овыпукления в $(d - 1)$ -пространстве плоскости \mathcal{L} .

Выход: $(d - 1)$ -грань $\mathcal{F} \subset \mathbb{R}^d$, лежащая в плоскости \mathcal{L} .

1. Пусть теперь \mathcal{V} — множество точек \mathcal{S} , лежащих в плоскости \mathcal{L} .
2. Если $|\mathcal{V}| = d$, то полученная грань симплицальна, дальнейшая обработка не требуется. Искомая грань — симплекс, построенный на d данных точках.
3. Иначе проецируем \mathcal{S} на \mathcal{L} используя аффинный базис (предварительно запоминая из какой точки $s \in \mathcal{S}$ получилась очередная точка $s' \in \mathcal{L}$) и рекурсивно строим выпуклую оболочку в аффинном подпространстве плоскости \mathcal{L} , если передано ребро \mathcal{E} , то проецируем его на \mathcal{L} и передаём в процедуру овыпукления в подпространстве.
4. Результатом овыпукления в подпространстве плоскости будет $(d - 1)$ -многогранник выраженный в терминах $(d - 1)$ -пространства, поэтому «поднимаем» его в d -мерное исходное пространство: очередную вершину s' подменяем исходной для неё точкой s . Производим аналогичную операцию для всех дочерних объектов данного многогранника.

3 Процесс заворачивания

Можно рассмотреть граф граней искомой выпуклой оболочки \mathcal{P} . Вершины графа сопоставляются с гранями \mathcal{P} . Две вершины являются соседними, если соответствующие им $(d - 1)$ -грани имеют общее $(d - 2)$ -ребро.

В начале процесса заворачивания нам известна какая-то вершина этого графа, соответствующая грани, построенной алгоритмом из предыдущего раздела. Также нам известны рёбра графа, выходящие из этой вершины, так как нам известны рёбра этой начальной грани.

В таком рассмотрении построение выпуклой оболочки соответствует обходу всех её граней, то есть обходу графа граней \mathcal{P} . Такой обход графа может быть осуществлён каким-либо поисковым алгоритмом. Наиболее компактную реализацию имеет алгоритм поиска в глубину. Эта реализация является рекурсивной.

Напомним, что один рекурсивный шаг поиска в глубину состоит в переборе всех не посещённых соседей текущей вершины и переходов в них с продолжением поиска оттуда. В геометрических терминах перебор соседей и переход в них соответствует перебору рёбер текущей грани и построению грани, соседней текущей через очередное рассматриваемое ребро. Такое построение осуществляется поворотом плоскости текущей грани вокруг

рассматриваемого ребра до касания какой-либо точки роя \mathcal{S} , не лежащей на рассматриваемом ребре. Такой поворот осуществляется аналогично шагу 5d алгоритма построения плоскости начальной грани.

3.1 Запуск поиска в глубину

Вход: $\mathcal{S} \subset \mathbb{R}^d$, возможна передача информации о начальной $(d - 1)$ -грани $\mathcal{F} \subset \mathbb{R}^d$.

Выход: d -многогранник $\mathcal{P} \subset \mathbb{R}^d$.

- Если не передали начальную грань:
 1. Выполняем построение начальной плоскости см. 2.1.
 2. Строим начальную грань \mathcal{F} см. 2.2.
- Для всех $(d - 2)$ -рёбер грани \mathcal{F} счётчик смежных граней устанавливаем равным единице.
- Запускаем поиск в глубину от грани \mathcal{F} .

3.2 Поиск в глубину

Имеется текущая грань \mathcal{F} .

1. Пока у грани \mathcal{F} есть рёбра со счётчиком, равным единице, повторяем:
 - (a) Перекатываемся через ребро со счётчиком равным единице.
 - (b) Всем рёбрам получившейся грани увеличиваем счётчик на один.
 - (c) Запускаем поиск в глубину от получившейся грани.
2. Завершаем процедуру.

3.3 Процедура перекатывания через ребро

Вход: $\mathcal{S} \in \mathbb{R}^d$, текущая $(d - 1)$ -грань \mathcal{F} и $(d - 2)$ -ребро \mathcal{E} текущей грани, через которое происходит перекатывание, $(o, \mathcal{B}_{\mathcal{F}})$ — аффинный базис плоскости содержащей грань \mathcal{F} . Вектор \vec{n} — нормаль грани \mathcal{F} .

Выход: Новая грань \mathcal{F} и внешняя нормаль к ней.

1. Построим аффинный базис ребра \mathcal{E} .
 - (a) Возьмём произвольную точку $o \in \mathcal{E}$ — начало аффинного базиса. Положим $\mathcal{B}_{\mathcal{E}} = \emptyset$.
 - (b) Пока $|\mathcal{B}_{\mathcal{E}}| < d - 2$: $\mathcal{B}_{\mathcal{E}} \leftarrow \mathcal{ON}(s_e - o, \mathcal{B}_{\mathcal{E}})$, где $s_e \in \mathcal{E}$.
2. Вычислим вектор \vec{v} , перпендикулярный оси вращения и лежащий в плоскости \mathcal{F} .
 - (a) Возьмём точку $f \in \mathcal{F} \setminus \mathcal{E}$.
 - (b) Тогда искомый вектор равняется $\vec{v} = \mathcal{ON}(f - o, \mathcal{B}_{\mathcal{E}})$.
3. Находим точку $s_* \in \mathcal{S}$ такую, что $s_* \notin \mathcal{F}$, и угол между \vec{v} и \vec{u} , наибольший среди всех точек роя (то есть скалярное произведение $\langle \vec{v}, \mathcal{N}(\vec{u}) \rangle$, наименьшее среди всех таких точек из \mathcal{S}), где $\vec{u} = \langle s - o, \vec{v} \rangle \vec{v} + \langle s - o, \vec{n} \rangle \vec{n}$ — вектор, перпендикулярный оси вращения и лежащий в новой плоскости, содержащей точку s_* . Соответствующий такой точке s_* вектор \vec{u} обозначим \vec{r} .

4. Аффинный базис плоскости $\mathcal{B}_{\mathcal{L}} \triangleq (o, \mathcal{B}_{\mathcal{L}} \cup \{\vec{r}\})$.
5. Вычислим нормаль \vec{n}_* новой плоскости: $\vec{n}_* = \langle \vec{r}, \vec{n} \rangle \vec{v} - \langle \vec{r}, \vec{v} \rangle \vec{n}$. При необходимости надо переориентировать \vec{n}_* так, чтобы $\langle \vec{n}_*, s - o \rangle < 0$, где s — точка роя, не лежащая в текущей плоскости \mathcal{F} , то есть такая, что $\langle \vec{n}_*, s - o \rangle \neq 0$.
6. Выполняем построение новой грани \mathcal{F}' на точках роя, попавших в плоскость \mathcal{L} , проходящую через ребро \mathcal{E} и точку s (см. процедуру 2.2).

3.4 Процедура получения базиса плоскости, содержащего базис ребра

Вход: базис $\mathcal{B}_{\mathcal{F}}$ $(d - 1)$ -грани \mathcal{F} , $(d - 2)$ -ребро \mathcal{E} этой грани (важен набор E точек, лежащих в этом ребре).

Выход: базис $\mathcal{B}_{\mathcal{F}}$, содержащий базис ребра \mathcal{E} .

1. Выбираем две точки p, p' из E , $p \neq p'$. Точку p полагаем началом аффинного базиса, нормированный вектор $p' - p$ полагаем первым вектором \vec{b}_1 конструируемого набора $\mathcal{B}'_{\mathcal{F}}$.
2. Берём точку $p'' \in \mathcal{F}$, $p'' \notin E$. Вектор $p'' - p$, нормированный на фоне \vec{b}_1 , полагаем $(d - 1)$ -м вектором \vec{b}_{d-1} конструируемого набора $\mathcal{B}'_{\mathcal{F}}$.
3. Для всех векторов $b \in \mathcal{B}_{\mathcal{F}}$ проверяем, является ли b линейно-независимым на фоне уже накопленного набора $\mathcal{B}'_{\mathcal{F}}$, и, если является, добавляем в $\mathcal{B}'_{\mathcal{F}}$ результат ортонормирования b на фоне текущего набора $\mathcal{B}'_{\mathcal{F}}$.

4 TODO

1. Перепланировать программу так, чтобы GW - конструктор объекта, из которого мы бы получали результат GW.
2. Разумно перекидать какие-то данные в глобальные переменные экземпляра класса: в частности овыпукляемый рой, строящиеся грани, строящаяся информация о соседстве граней.
3. При построении очередной грани выкидывать из роя те точки, которые были в плоскости, но не стали вершинами.