On convex polyhedron computations using floating point arithmetic

Andreas Löhne

Friedrich Schiller University Jena, Germany

WU Wien, October, 7, 2022

Polyhedron: $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$

Polytope: bounded polyhedron

Many aspects for polyhedra can be formulated in terms of polytopes:

If P is a line-free polyhedron in $\mathbb{R}^d$, then $\operatorname{cl}\operatorname{cone}(P \times \{1\})$ is a line-free polyhedral cone in $\mathbb{R}^{d+1}$, which is given by a bounded base B, which is a polytope in $\mathbb{R}^d$.

Vertex Enumeration and Convex Hull Problem

Vertex Enumeration: H-represented polytope P with $0 \in \operatorname{int} P$:

$$P = \{x \in \mathbb{R}^d \mid Ax \leq \mathbb{1}\}.$$

Goal: compute a V-representation:

$$P = \operatorname{conv}\{v_1, \ldots, v_k\}$$

Convex hull problem: V-represented polytope P with $0 \in \operatorname{int} P$:

$$P = \operatorname{conv}\{v_1, \ldots, v_k\}$$

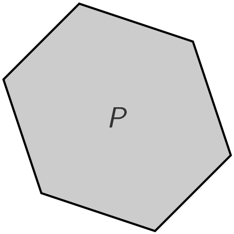
Goal: compute a H-representation:

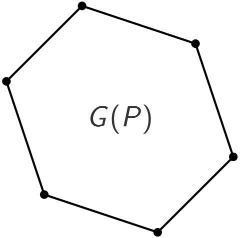
$$P = \{x \in \mathbb{R}^d \mid Ax \leq \mathbb{1}\}.$$

Both problems are equivalent by polarity.

Properties of 2-polytopes P:

- $v = e$
- The (vertex-edge) graph is a cycle

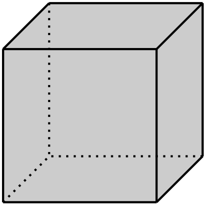


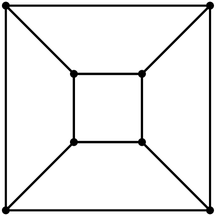


$\rightarrow$ very simple structure

Properties of 3-polytopes:

- $v - e + f = 2$ (Euler's formula)
- The (vertex-edge) graph is planar and 3-connected





More properties of 3-polytopes

Steinitz's theorem: A graph G is the vertex-edge graph of a 3-polytope if and only if G is planar and 3-connected.

- Ernst Steinitz (1871 – 1928)
- *"the most important and deepest known result on 3-polytopes"* (Branko Grünbaum)
- One of many consequences: Every 3-polytope can be realized by integer coordinates
- No similar result for higher dimensions !

Theorem: (follows from Steinitz's th., e.g. [Richter-Gebert])

The realization space of a 3-polytope is an open ball (of dimension $e-6$)

Example: cube, one realization has coordinates

More properties of 3-polytopes

Theorem: (follows from Steinitz's th., e.g. [Richter-Gebert])

The realization space of a 3-polytope is an open ball (of dimension $e - 6$)

Example: cube, one realization has coordinates

$$\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}, \begin{pmatrix}1\\1\\0\end{pmatrix}, \begin{pmatrix}1\\0\\1\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}, \begin{pmatrix}1\\1\\1\end{pmatrix}.$$

More properties of 3-polytopes

Theorem: (follows from Steinitz's th., e.g. [Richter-Gebert])

The realization space of a 3-polytope is an open ball (of dimension $e-6$)

Example: cube, one realization has coordinates

$$\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}, \begin{pmatrix}1\\1\\0\end{pmatrix}, \begin{pmatrix}1\\0\\1\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}, \begin{pmatrix}1\\1\\1\end{pmatrix}.$$

fix a base (as affine transformations are not of interest)

Theorem: (follows from Steinitz's th., e.g. [Richter-Gebert])

The realization space of a 3-polytope is an open ball (of dimension $e-6$)

Example: cube, one realization has coordinates

$$\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}, \begin{pmatrix}1\\1\\0\end{pmatrix}, \begin{pmatrix}1\\0\\1\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}, \begin{pmatrix}1\\1\\1\end{pmatrix}.$$

fix a base (as affine transformations are not of interest)

fixed to maintain combinatorics

Theorem: (follows from Steinitz's th., e.g. [Richter-Gebert])

The realization space of a 3-polytope is an open ball (of dimension $e - 6$)

Example: cube, one realization has coordinates

$$\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}, \begin{pmatrix}z_1\\z_2\\0\end{pmatrix}, \begin{pmatrix}z_3\\0\\z_4\end{pmatrix}, \begin{pmatrix}0\\z_5\\z_6\end{pmatrix}, \begin{pmatrix}?\\?\\?\end{pmatrix}.$$

fix a base (as affine transformations are not of interest)

fixe these entries to maintain combinatorics

variable, to maintain convexity, we need
$z_1 + z_2 > 1,\ z_3 + z_4 > 1,\ z_5 + z_6 > 1,\ z_1 > 0, \ldots, z_6 > 0$

Theorem: (follows from Steinitz's th., e.g. [Richter-Gebert])

The realization space of a 3-polytope is an open ball (of dimension $e-6$)

Example: cube, one realization has coordinates

$$\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}, \begin{pmatrix}z_1\\z_2\\0\end{pmatrix}, \begin{pmatrix}z_3\\0\\z_4\end{pmatrix}, \begin{pmatrix}0\\z_5\\z_6\end{pmatrix}, \begin{pmatrix}y_1\\y_2\\y_3\end{pmatrix}.$$

fix a base (as affine transformations are not of interest)

fixe these entries to maintain combinatorics

variable, to maintain convexity, we need
$z_1 + z_2 > 1,\ z_3 + z_4 > 1,\ z_5 + z_6 > 1,\ z_1 > 0, \ldots, z_6 > 0$

fixed by the choice of $z_1, \ldots, z_6$

Universality theorem for 4-polytopes (Richter-Gebert, 1994):
For every primary semi-algebraic set V defined over $\mathbb{Z}$ there is a 4-polytope whose realization space is stably equivalent to V.

Some consequences:

- All algebraic numbers are needed to coordinatize all 4-polytopes.
- The realizability problem for 4-polytopes is NP-hard.

4-polytopes are much more complicated than 3-polytopes

Exact rational vs. Floating Point Arithmetic

[Avis; Bremner, Seidel (1997)]

How good are convex hull algorithms?

Contra exact rational arithmetic:

"In examples where input numbers are very large such as the products of cyclic polytopes, cddr+ was thousands of times slower than cddf+ on some inputs."

Contra floating point arithmetic:

"The convex hull problem has the nice property that it is possible to perform all computations in exact rational arithmetic; this is especially desirable in applications such as combinatorial optimization where an exact answer is desired rather than just an approximation."

Do we really obtain an "approximation"?

Other software using floating point arithmetic

Quickhull package: http://quickhull.org

[Barber, Dobkin, Huhdanpaa, 1996]
The Quickhull Algorithm for Convex Hulls

"In $\mathbb{R}^2$, there are several robust convex hull ... algorithms [Fortune 1989; Guibas et al. 1993; Li and Milenkovic 1990]."

"In $\mathbb{R}^3$, Sugihara [1992] and Dey et al. [1992] produce a topologically robust convex hull and Delaunay triangulation. ... The output may contain unbounded geometric faults."

"We have implemented Quickhull for general dimension."

An attempt ...

Cornell University

arXiv > math > arXiv:2007.06325

Mathematics > Optimization and Control

[Submitted on 13 Jul 2020 (v1), last revised 4 Jul 2022 (this version, v3)]

Approximate Vertex Enumeration

Andreas Löhne

Given: $P = \{x \in \mathbb{R}^d \mid Ax \leq \mathbb{1}\}$

For $\varepsilon \geq 0$, define: $(1+\varepsilon)P := \{x \in \mathbb{R}^d \mid Ax \leq (1+\varepsilon)\mathbb{1}\}$.

Goal: construct iteratively an ε-*approximate V-representation* of P, i.e. $V = \{v_1, \ldots, v_k\} \subseteq \mathbb{R}^d$ such that

$$P \subseteq \operatorname{conv} V \subseteq (1+\varepsilon)P.$$

Remark 1: For $\varepsilon = 0$ we obtain a V-represention of P.

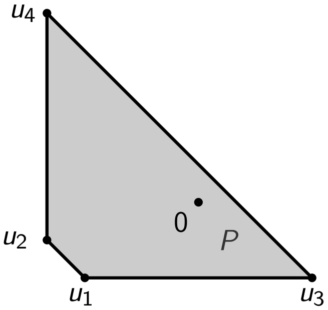
Remark 2: P and $\operatorname{conv} V$ are **not** required to be combinatorially equivalent.

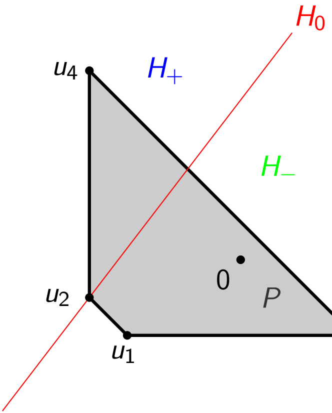
Iterative scheme for vertex enumeration:

Init: simplex, H- and V-representation known
Iteration: add one inequality and update V-representation

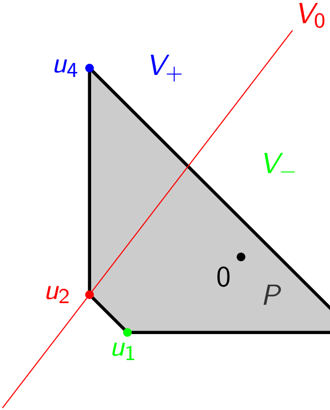
$V_0 \leftarrow V \cap \{x \in \mathbb{R}^d \mid h^T x = 1\}$
$V_+ \leftarrow V \cap \{x \in \mathbb{R}^d \mid h^T x > 1\}$
$V_- \leftarrow V \cap \{x \in \mathbb{R}^d \mid h^T x < 1\}$
foreach $(v_1, v_2) \in V_- \times V_+$ **do**
    if $|J_=(v_1) \cap J_=(v_2)| \geq d - 1$ **then**
        compute $v \in \mathrm{conv}\{v_1, v_2\} \cap H_0$
        $V \leftarrow V \cup \{v\}$
    end
end
$V' \leftarrow V \setminus V_+$

Double description method (DDM)

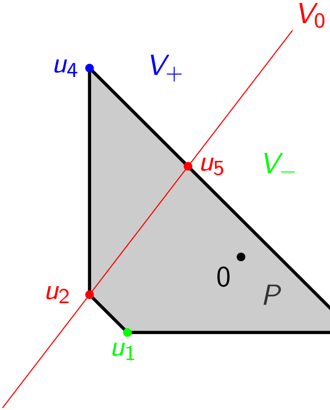




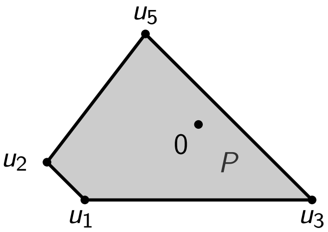
H_0
u_4
H_+
H_-
0
P
u_2
u_1
u_3



V_0
u_4
V_+
V_-
0
P
u_2
u_1
u_3


V_0
u_4
V_+
u_5
V_-
0
u_2
P
u_1
u_3

Double description method (DDM)



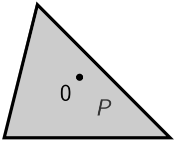
Approximate double description method (ADDM)

Iterative scheme for approximate vertex enumeration:

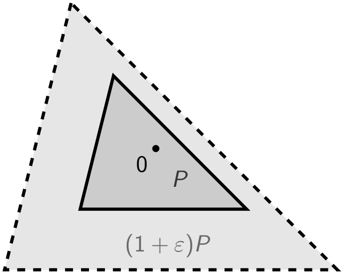
Init: simplex, H- and V-representation known
Iteration: add one inequality and update V-representation

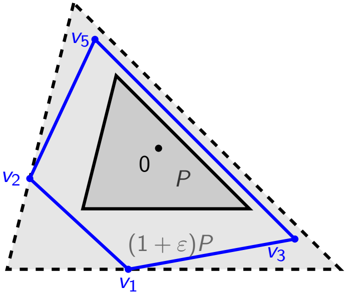
$V_0 \leftarrow V \cap \{x \in \mathbb{R}^d \mid 1 \leq h^T x \leq 1 + \varepsilon\}$
$V_+ \leftarrow V \cap \{x \in \mathbb{R}^d \mid h^T x > 1 + \varepsilon\}$
$V_- \leftarrow V \cap \{x \in \mathbb{R}^d \mid h^T x < 1\}$
foreach $(v_1, v_2) \in V_- \times V_+$ **do**
    if $|J_{\geq}(v_1) \cap J_{\geq}(v_2)| \geq d - 1$ **then**
        compute $v \in \mathrm{conv}\{v_1, v_2\} \cap H_0$
        $V \leftarrow V \cup \{v\}$
    end
end
$V' \leftarrow V \setminus V_+$

Example



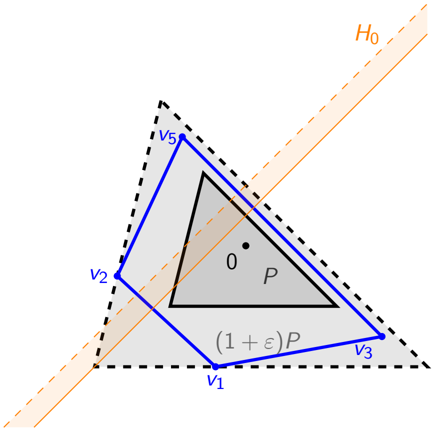
Example



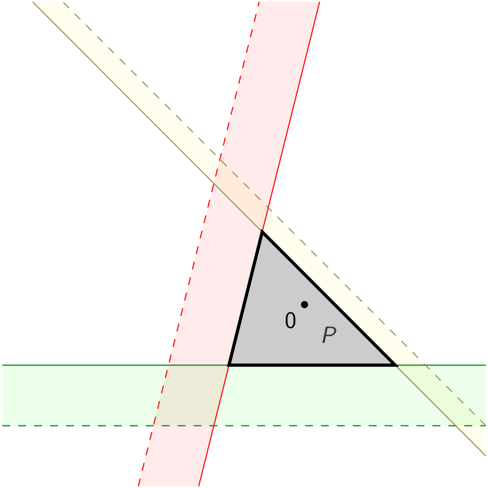


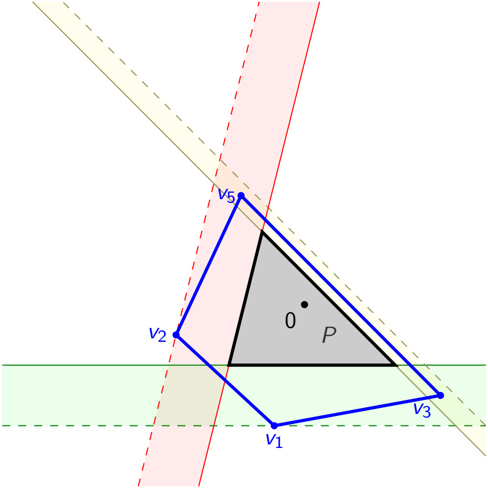
v_5
0
P
v_2
$(1+\varepsilon)P$
v_3
v_1

Example

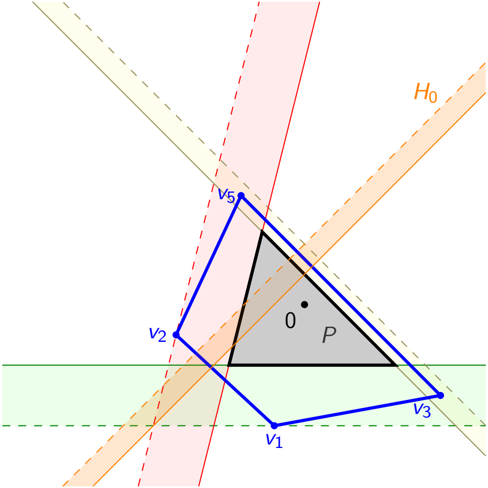


Example

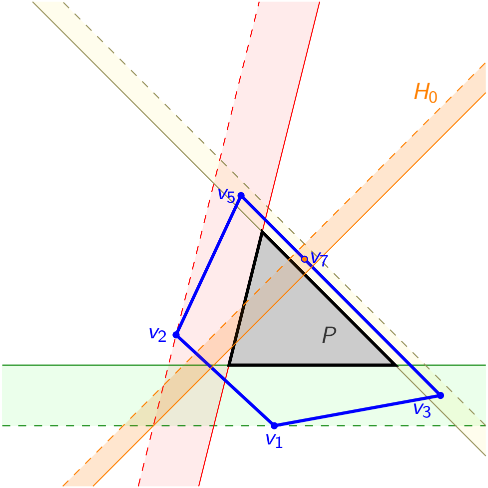



v_5
0
P
v_2
v_3
v_1

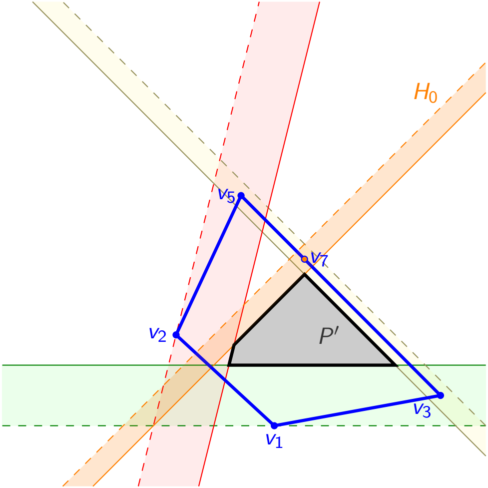
Example



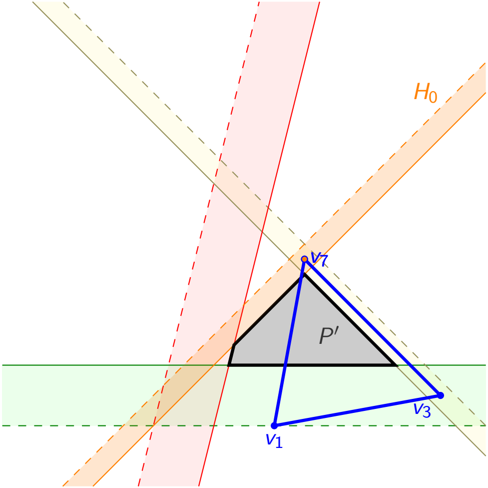
Example



Example

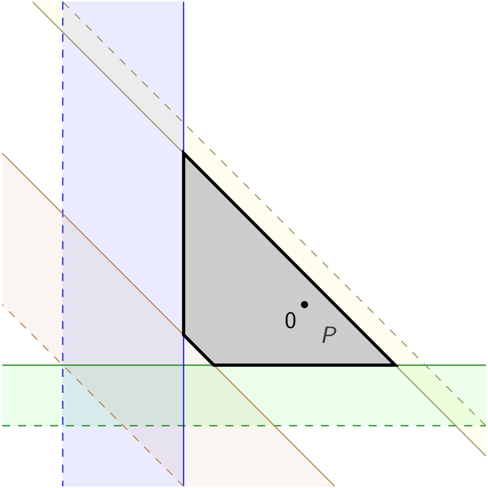


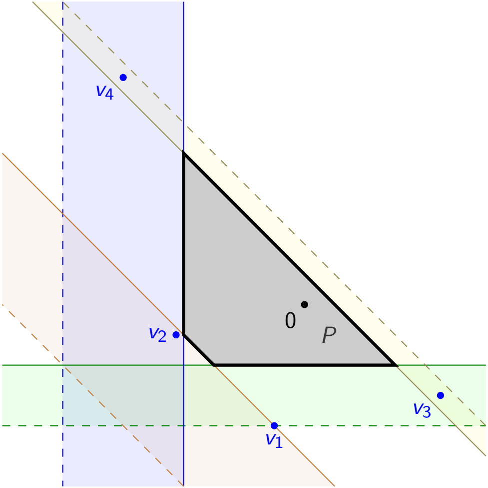
Example



Is ADDM correct?

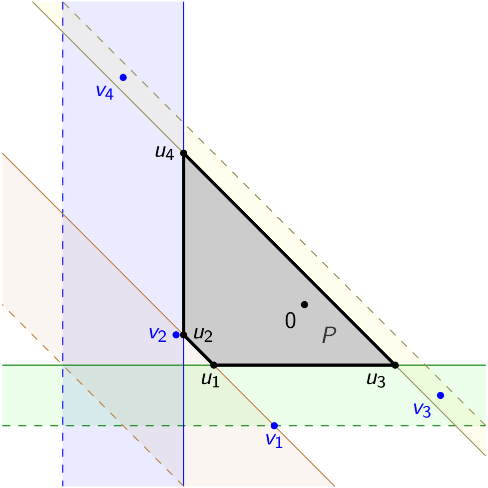
- Yes, for $d = 2$.
- Yes, for $d = 3$.
- Open, for $d \geq 4$.

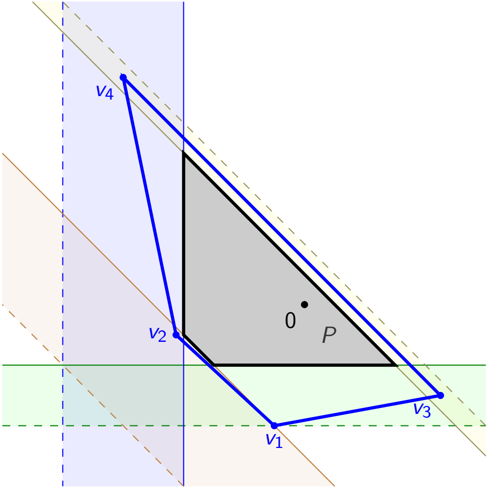

0
P



v_4
v_2
0
P
v_3
v_1

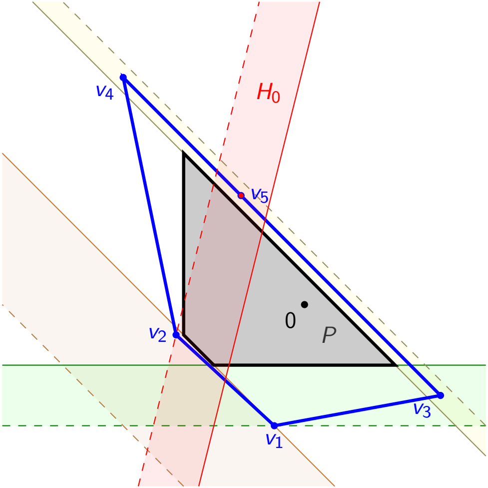
Example

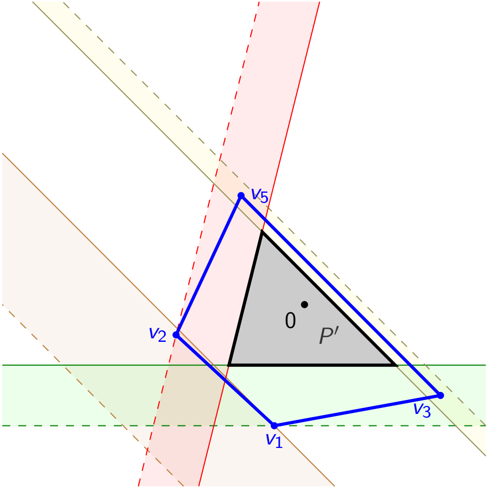




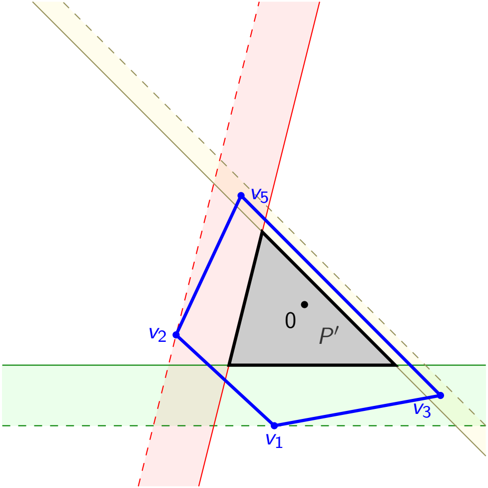
v_4
v_2
0
P
v_3
v_1

Example

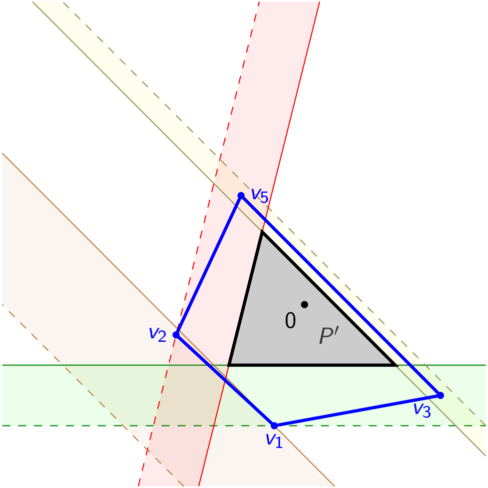



v_5
0
P'
v_2
v_3
v_1

Example



Example



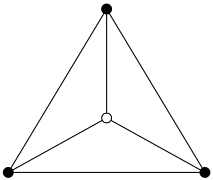
How to prove correctness for $d \in \{2, 3\}$?

Why does the proof only works for $d \in \{2, 3\}$?

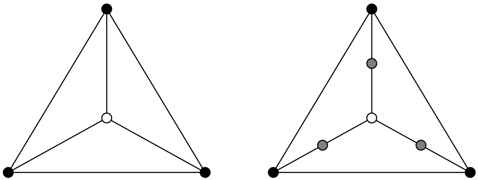
Sketch:

- second algorithm for approximate VE: *graph algorithm*
- prove correctness of graph algorithm
- show that ADDM computes a superset of vertices computed by graph algorithm

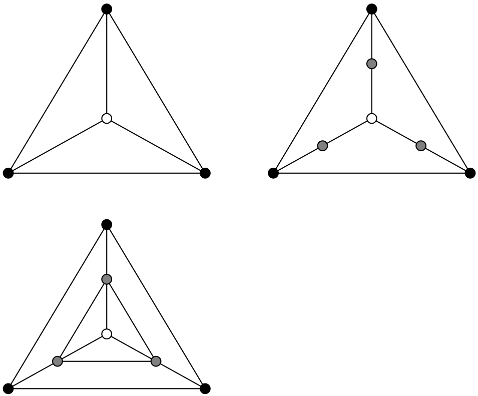
Iteration 1



Iteration 1

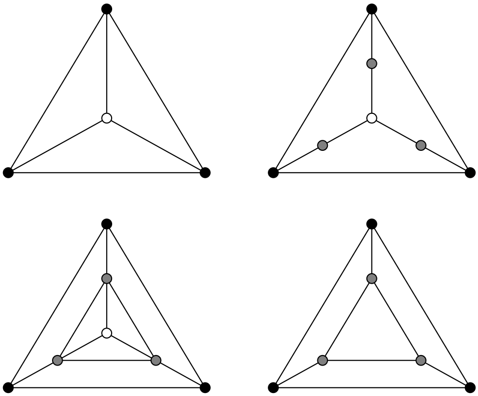


Iteration 1



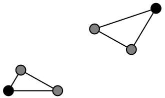
Core of the graph algorithm ($d = 3$)

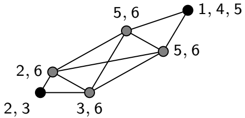
Iteration 1



graph algorithm

approximate DDM





Theorem: The graph algorithm computes a subgraph of the graph computed by ADDM.

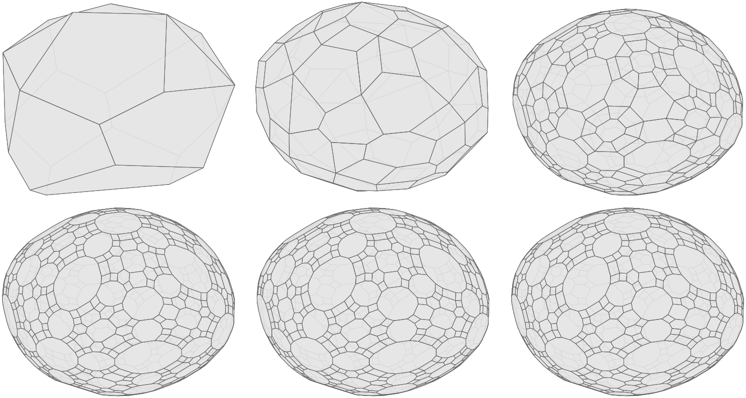
E ... error of coordinates caused by using imprecise arithmetic (difficult to quantify)

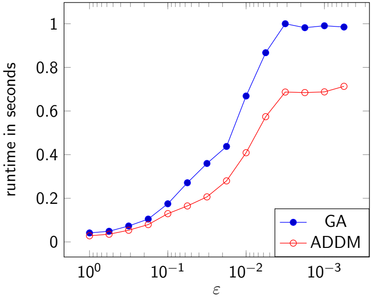
$\delta > 0$ ... radius of ball around origin contained in P

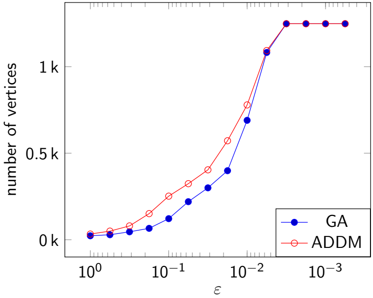
Theorem: Variants of both algorithms remain correct if imprecise arithmetic is used and:

$$E \leq \frac{\varepsilon\delta}{4}.$$

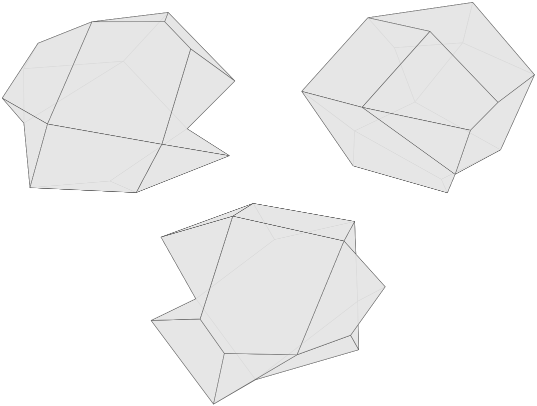
Results of the **graph algorithm** for
$\varepsilon \in \{10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$



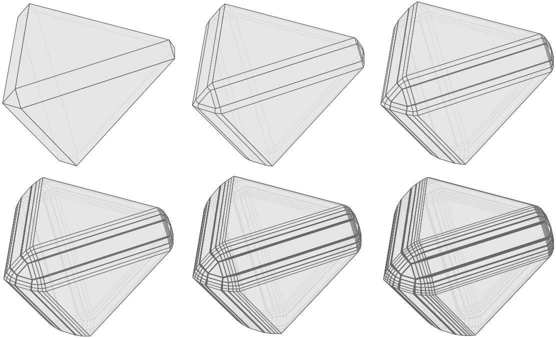

runtime in seconds
1
0.8
0.6
0.4
0.2
0
10^0
10^{-1}
10^{-2}
10^{-3}
ε
GA
ADDM


number of vertices
1 k
0.5 k
0 k
10^0
10^{-1}
10^{-2}
10^{-3}
ε
GA
ADDM

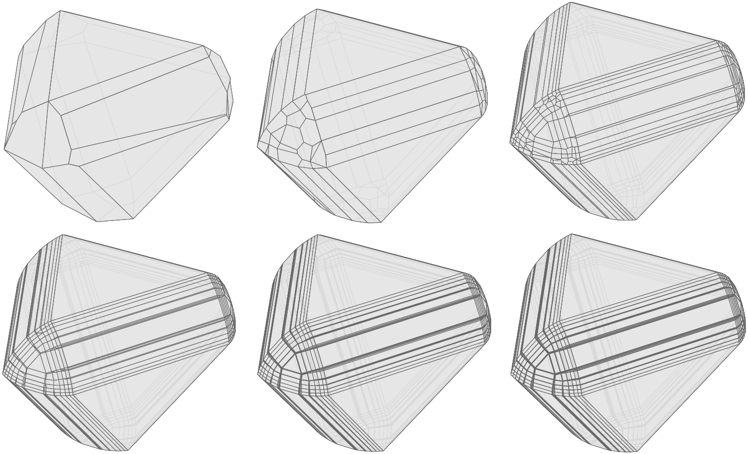
The **graph algorithm** produces **non-convex objects**; above example for $\varepsilon = 2$, different viewpoints

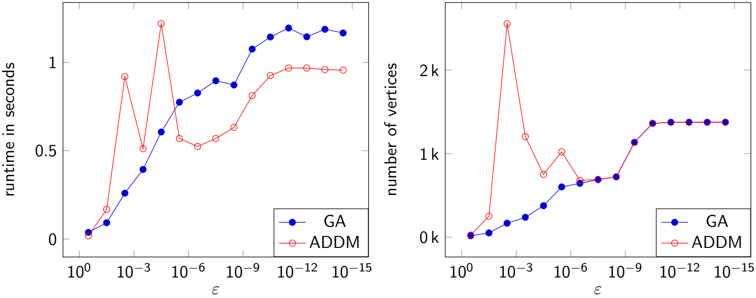


Example: P_0 regular simplex in $\mathbb{R}^3$, edge length 1 symmetrically placed around origin. P_i is defined as Minkowski sum of P_{i-1} and the polar of P_{i-1}.
Pictures: $P_1, \ldots, P_6$

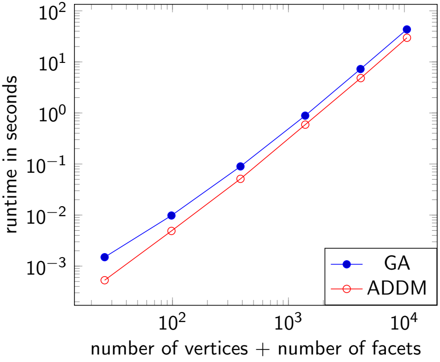


Results of graph algorithm: P_6 for $\varepsilon \in \{10^{-1}, \ldots, 10^{-6}\}$




runtime in seconds
1
0.5
0
GA
ADDM
10^0 10^{-3} 10^{-6} 10^{-9} 10^{-12} 10^{-15}
ε
number of vertices
2 k
1 k
0 k
GA
ADDM
10^0 10^{-3} 10^{-6} 10^{-9} 10^{-12} 10^{-15}
ε

above example for $\varepsilon = 10^{-9}$



Conclusions and open problems

Conclusions:

- Floating point implementations for vertex enumeration / convex hull problem can produce wrong results
- We introduced the approximate vertex enumeration problem and two solution methods
- Both methods were shown to be correct for $d = 2, 3$ for floating point arithmetic if the imprecision is not too high.

Open Problems:

- Is ADDM correct for $d \geq 4$?
- Is there any other practically relevant correct method for the approximate vertex enumeration problem for $d \geq 4$?
- Do floating point implementations of vertex enumeration methods fail in practice? In particular for $d \geq 4$?