An output-sensitive algorithm for computing projections of resultant polytopes

Christos Konaxis

Joint work with I.Z. Emiris. V. Fisikopoulos, and L. Peñaranda

Archimedes Center for Modeling, Analysis and Computation,
Univeristy of Crete

ACMAC seminar, March 8, 2012

Outline of the talk

1. Introduction
2. Resultant polytopes
3. The algorithm

The problem

Given polynomials in $K[x, y]$:

$$f_0 = u_2 x + u_1 y + u_0$$
$$f_1 = c_{13} xy + c_{12} x + c_{11} y + c_{10}$$
$$f_2 = c_{23} xy^2 + c_{22} xy + c_{21} x + c_{20}$$

what is the condition that their coefficients $(c_{10}, \ldots, u_2)$ must satisfy so that the system $f_0 = f_1 = f_2 = 0$ has a solution in $(\mathbb{C}^*)^2$?

$$-c_{20} c_{21} u_1^3 c_{12} c_{13}^2 - u_0 c_{21}^2 u_1^2 c_{11} c_{13}^2 + c_{21}^2 u_1^3 c_{10} c_{13}^2 + \ldots \text{another 57 terms} \ldots = 0$$

The problem

When some of the polynomials' coefficients are specialized to constants,

$$f_0 = u_2 x + u_1 y + u_0$$
$$f_1 = 3xy + 11x + 4y + 1$$
$$f_2 = 2xy^2 + 5xy + x + 3$$

what is the condition that the remaining symbolic coefficients must satisfy so that the system $f_0 = f_1 = f_2 = 0$ has a solution in $(\mathbb{C}^*)^2$?

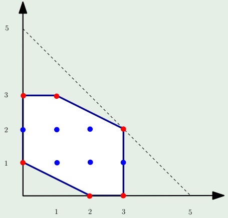
$$80u_0^3 - 136u_1u_0^2 - 180u_2u_0^2 + 288u_1^2u_0 - 674u_2u_1u_0 + 910u_2^2u_0 - 8u_1^3 - 62u_2u_1^2 + 183u_2^2u_1 - 250u_2^3 = 0$$

Preliminaries

The support $A(f)$ of a polynomial f is the set of exponents of its monomials with non-zero coefficient. The Newton polytope $N(f)$ of f is the convex hull of its support.

Example

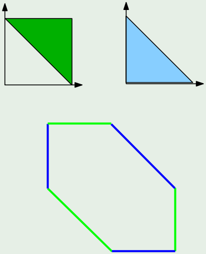
$$f(x,y) = 8y + xy - 24y^2 - 16x^2 + 220x^2y - 34xy^2 - 84x^3y + 6x^2y^2 - 8xy^3 + 8x^3y^2 + 8x^3 + 18y^3$$

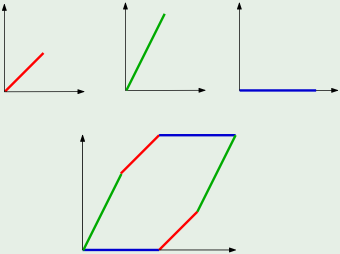


The Newton polytope is the equivalent notion of total degree in sparse elimination theory.

The Minkowski sum of polytopes $P_0, \dots, P_n$ (resp., supports $A_0, \dots, A_n$) is the polytope $P = P_0 + \cdots + P_n := \{p_0 + \cdots + p_n \mid p_i \in P_i\}$
(resp., $P = \mathrm{conv}(A_0 + \cdots + A_n) := \mathrm{conv}\{a_0 + \cdots + a_n \mid a_i \in A_i\}$).

Example



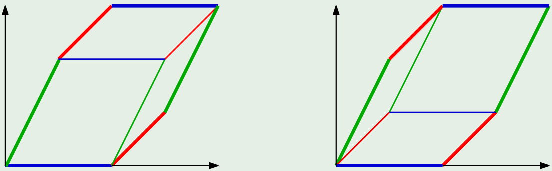


Mixed subdivisions

A fine mixed subdivision of $P = \mathrm{conv}(A_0 + \cdots + A_n) \subset \mathbb{R}^n$, is any family of cells partitioning P and intersecting properly as Minkowski sums of faces $F_i \subset \mathrm{conv}(A_i)$, where $\dim(\sum_i F_i) = \sum_i \dim F_i$.

Example

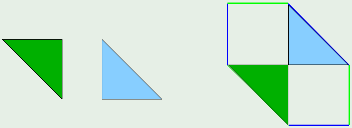
$$f_0 := c_{00} - c_{01}x_1x_2, \; f_1 := c_{10} - c_{11}x_1x_2^2, \; f_2 := c_{20} - c_{21}x_1^2$$

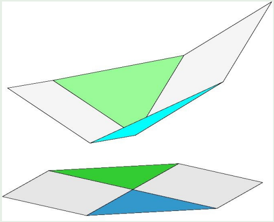


Regular mixed subdivisions

A regular mixed subdivision of $P = \text{conv}(A_0 + \cdots + A_n) \subset \mathbb{R}^n$ is obtained as the projection of the lower (or upper) hull of the Minkowski sum of lifted polytopes $\hat{P}_i := \text{conv}\{(a_i, \omega_i(a_i)) \mid a_i \in A_i\} \subset \mathbb{R}^{n+1}$. Generic ω_i produce regular fine mixed subdivisions.

Example





The sparse (toric) resultant

Definition

The sparse resultant $\mathcal{R}$ of polynomials $f_0, \ldots, f_n$, where

$$f_i = \sum_{a \in A_i} c_{ia} x^a, \quad x^a = (x_1^{a_1}, \ldots, x_n^{a_n}),$$

with symbolic c_{ia} and exponents in $A_i \subset \mathbb{Z}^n$, is the unique irreducible integer polynomial in the c_{ij}'s, which vanishes iff the f_i have a common complex root.

Definition

The sparse resultant $\mathcal{R}$ of polynomials $f_0, \ldots, f_n$, where

$$f_i = \sum_{a \in A_i} c_{ia} x^a, \quad x^a = (x_1^{a_1}, \ldots, x_n^{a_n}),$$

with symbolic c_{ia} and exponents in $A_i \subset \mathbb{Z}^n$, is the unique irreducible integer polynomial in the c_{ij}'s, which vanishes iff the f_i have a common complex root.

Example

Let $A_0 = A_1 = \{0, 1\}$. Then $f_0 = c_{01}x + c_{02}$, $f_1 = c_{11}x + c_{12}$.

$$\mathcal{R} = c_{01}c_{12} - c_{02}c_{11} \in \mathbb{Z}[c_{01}, c_{02}, c_{11}, c_{12}]$$

The sparse (toric) resultant

Definition

The sparse resultant $\mathcal{R}$ of polynomials $f_0, \ldots, f_n$, where

$$f_i = \sum_{a \in A_i} c_{ia} x^a, \quad x^a = (x_1^{a_1}, \ldots, x_n^{a_n}),$$

with symbolic c_{ia} and exponents in $A_i \subset \mathbb{Z}^n$, is the unique irreducible integer polynomial in the c_{ij}'s, which vanishes iff the f_i have a common complex root.

Example

Let $A_0 = A_1 = \{0, 1\}$. Then $f_0 = c_{01}x + c_{02}, \; f_1 = c_{11}x + c_{12}$.

$$\mathcal{R} = c_{01}c_{12} - c_{02}c_{11} \in \mathbb{Z}[c_{01}, c_{02}, c_{11}, c_{12}]$$

The Newton polytope $N(\mathcal{R})$ of the resultant is called the resultant polytope.

Resultant polytope

Theorem (Sturmfels'94)

There exists a surjection ρ from the regular fine mixed subdivisions of $P = conv(A_0 + \cdots + A_n)$ onto the vertices of $N(\mathcal{R})$. Subdivisions S, S' are equivalent iff $\rho_S = \rho_{S'}$.

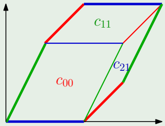
Resultant polytope

Theorem (Sturmfels'94)

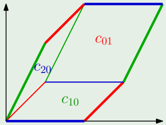
There exists a surjection ρ from the regular fine mixed subdivisions of $P = conv(A_0 + \cdots + A_n)$ onto the vertices of $N(\mathcal{R})$. Subdivisions S, S' are equivalent iff $\rho_S = \rho_{S'}$.

Example

$$f_0 := c_{00} - c_{01}st, \quad f_1 := c_{10} - c_{11}st^2, \quad f_2 := c_{20} - c_{21}s^2$$



$c_{00}^4 c_{11}^2 c_{21}$



$c_{01}^4 c_{10}^2 c_{20}$

$$\rho_S = (4, 0, 0, 2, 0, 1)$$
$$\rho_{S'} = (0, 4, 2, 0, 1, 0)$$

The Cayley Trick

Theorem

Regular mixed subdivisions of $P = conv(A_0 + \cdots + A_n) \subset \mathbb{R}^n$ *correspond bijectively to regular triangulations of* $\mathcal{A} := \bigcup_{i=0}^{n}(A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$*:*
e_i *form an affine basis of* $\mathbb{R}^n$.

The Cayley Trick

Theorem

Regular mixed subdivisions of $P = conv(A_0 + \cdots + A_n) \subset \mathbb{R}^n$ *correspond bijectively to regular triangulations of* $\mathcal{A} := \bigcup_{i=0}^{n}(A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$*:*
e_i *form an affine basis of* $\mathbb{R}^n$.

$$\begin{aligned} f_0 &= c_{01}x^2 + c_{02}y^2 + c_{03}x^2y^2 \\ f_1 &= c_{11}x^2 + c_{12}y^2 + c_{14} \end{aligned}$$

Theorem

Regular mixed subdivisions of $P = conv(A_0 + \cdots + A_n) \subset \mathbb{R}^n$ *correspond bijectively to regular triangulations of* $\mathcal{A} := \bigcup_{i=0}^{n}(A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$*:*
e_i *form an affine basis of* $\mathbb{R}^n$.

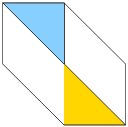
$$f_0 = c_{01}x^2 + c_{02}y^2 + c_{03}x^2y^2$$
$$f_1 = c_{11}x^2 + c_{12}y^2 + c_{14}$$

The Cayley Trick

Theorem

Regular mixed subdivisions of $P = conv(A_0 + \cdots + A_n) \subset \mathbb{R}^n$ correspond bijectively to regular triangulations of $\mathcal{A} := \bigcup_{i=0}^{n}(A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$: e_i form an affine basis of $\mathbb{R}^n$.

$$
\begin{aligned}
f_0 &= c_{01}x^2 + c_{02}y^2 + c_{03}x^2y^2 \\
f_1 &= c_{11}x^2 + c_{12}y^2 + c_{14}
\end{aligned}
$$

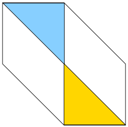


Theorem

Regular mixed subdivisions of $P = conv(A_0 + \cdots + A_n) \subset \mathbb{R}^n$ *correspond bijectively to regular triangulations of* $\mathcal{A} := \bigcup_{i=0}^{n} (A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$*:*
e_i *form an affine basis of* $\mathbb{R}^n$.

$$f_0 = c_{01}x^2 + c_{02}y^2 + c_{03}x^2y^2$$
$$f_1 = c_{11}x^2 + c_{12}y^2 + c_{14}$$

$$\mathcal{A} := \{(2,0,0),(0,2,0),(2,2,0),$$

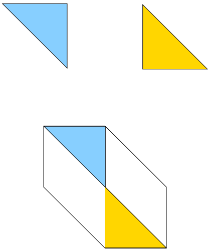


Theorem

Regular mixed subdivisions of $P = conv(A_0 + \cdots + A_n) \subset \mathbb{R}^n$ correspond bijectively to regular triangulations of $\mathcal{A} := \bigcup_{i=0}^{n}(A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$: e_i form an affine basis of $\mathbb{R}^n$.

$$f_0 = c_{01}x^2 + c_{02}y^2 + c_{03}x^2y^2$$
$$f_1 = c_{11}x^2 + c_{12}y^2 + c_{14}$$

$$\mathcal{A} := \{(2,0,0),(0,2,0),(2,2,0),(2,0,1),(0,2,1),(0,0,1)\}$$

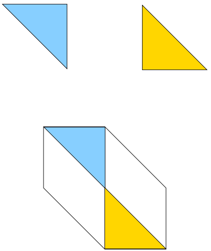


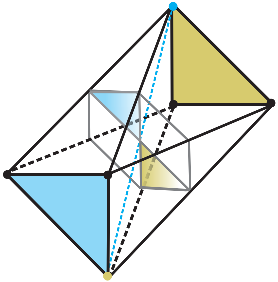
Theorem

Regular mixed subdivisions of $P = conv(A_0 + \cdots + A_n) \subset \mathbb{R}^n$ correspond bijectively to regular triangulations of $\mathcal{A} := \bigcup_{i=0}^{n}(A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$: e_i form an affine basis of $\mathbb{R}^n$.

$$f_0 = c_{01}x^2 + c_{02}y^2 + c_{03}x^2y^2$$
$$f_1 = c_{11}x^2 + c_{12}y^2 + c_{14}$$

$$\mathcal{A} := \{(2,0,0),(0,2,0),(2,2,0),(2,0,1),(0,2,1),(0,0,1)\}$$

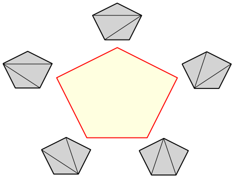




Theorem (Gelfand, Kapranov, Zelevinsky)

For every point set $\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\} \subset \mathbb{R}^d$, *there exists a polytope* $\Sigma(\mathcal{A}) \subset \mathbb{R}^{|\mathcal{A}|}$, *the* *secondary polytope* *of* $\mathcal{A}$, *s.t.*

1. *Vertices of* $\Sigma(\mathcal{A})$ *correspond bijectively to regular triangulations of* $\mathcal{A}$ *via*
$$T \mapsto \varphi_T : \ (\varphi_T)_i = \sum_{a_i \in \mathrm{Vert}(\sigma):\sigma \in T} \mathrm{Vol}(\sigma), \ i = 1, \dots, |\mathcal{A}|,$$
2. *Edges correspond to flips.*

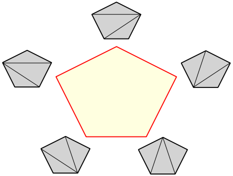


Secondary polytope of a pentagon.

Theorem (Gelfand, Kapranov, Zelevinsky)

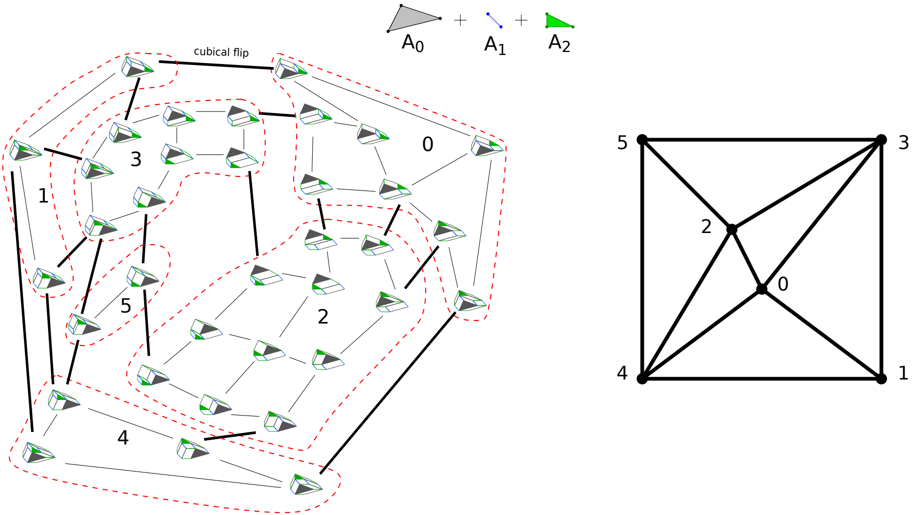
For every point set $\mathcal{A} = \{a_1, \ldots, a_{|\mathcal{A}|}\} \subset \mathbb{R}^d$*, there exists a polytope* $\Sigma(\mathcal{A}) \subset \mathbb{R}^{|\mathcal{A}|}$*, the* ***secondary polytope*** *of* $\mathcal{A}$*, s.t.*

1. *Vertices of* $\Sigma(\mathcal{A})$ *correspond bijectively to regular triangulations of* $\mathcal{A}$ *via*
$$T \mapsto \varphi_T : \ (\varphi_T)_i = \sum_{a_i \in \mathrm{Vert}(\sigma) : \sigma \in T} \mathrm{Vol}(\sigma), \ i = 1, \ldots, |\mathcal{A}|,$$
2. *Edges correspond to flips.*



Secondary polytope of a pentagon.

$N(\mathcal{R})$ is a Minkowski summand of $\Sigma(\mathcal{A})$.



(Top) Input polytopes. (Left) Graph of mixed subdivisions of A_0, A_1, A_2 (equiv., triangulations of $\mathcal{A}$): the 6 equivalence classes are dotted. (Right) $N(\mathcal{R}) \subset \mathbb{R}^3$.

The problem

Given $n + 1$ polynomials in n variables with fixed supports $A_i \subset \mathbb{Z}^n$, efficiently compute $\pi(N(\mathcal{R})) = \mathit{\Pi} \subset \mathbb{R}^m$ for some orthogonal projection π.

Idea

- Avoid computing the whole secondary polytope $\Sigma(\mathcal{A})$.
- Incrementally construct $\mathit{\Pi}$ using an oracle that given a direction produces a vertex of $\mathit{\Pi}$, extremal wrt this direction.

Interpolation of resultants

$$f_0 = a_2x^2 + a_1x + a_0,\ \ f_1 = b_1x^2 + b_0$$

$N(\mathcal{R})$: convex hull of $(0,2,0,1,1)$, $(0,0,2,2,0)$, $(2,0,0,0,2)$.
Contains 1 additional lattice point: $(1,0,1,1,1)$.
Possible monomials of $\mathcal{R}$: $a_1^2b_1b_0,\ \ a_0^2b_1^2,\ \ a_2a_0b_1b_0,\ \ a_2^2b_0^2$

Using the 4 monomials and the Horn-Kapranov parameterization of $\mathcal{R}$:

$$a_2 = (2t_1 + t_2)t_3^2t_4,\ \ a_1 = (-2t_1 - 2t_2)t_3t_4,\ a_0 = t_2t_4,\ \ b_1 = -t_1t_3^2t_5,\ \ b_0 = t_1t_5,$$

construct a 4×4 matrix whose kernel vector $(1,1,-2,1)$ contains their coefficients:

$$\mathcal{R} = a_1^2b_1b_0 + a_0^2b_1^2 - 2a_2a_0b_1b_0 + a_2^2b_0^2$$

Implicitization of parametric (hyper)surfaces

$$(y_1, y_2, y_3) = (x_1x_2,\ x_1x_2^2,\ x_1^2).$$

$$f_0 := c_{00} - c_{01}x_1x_2,\ \ f_1 := c_{10} - c_{11}x_1x_2^2,\ \ f_2 := c_{20} - c_{21}x_1^2$$

$$\mathcal{R} = -c_{00}^4c_{11}^2c_{21} + c_{01}^4c_{10}^2c_{20},\ \ N(\mathcal{R}) = ((4,0,0,2,0,1),(0,4,2,0,1,0)) \subset \mathbb{R}^6$$

$$\text{Define } \pi:\ \ \pi(c_{00}, c_{01}, c_{10}, c_{11}, c_{20}, c_{21}) = (y_1, -1, y_2, -1, y_3, -1)$$

Then, $\pi(\mathcal{R}) = -y_1^4 + y_2^2y_3$ is the implicit equation and
$\Pi = \pi(N(\mathcal{R})) = ((4,0,0),(0,2,1))$ is the implicit polytope.

Note that here: $\pi(N(\mathcal{R})) = N(\pi(\mathcal{R}))$

Existing work

Theory

- Michiels & Verschelde [DCG'99] define equivalence classes of mixed subdivisions mapping onto $N(\mathcal{R})$ vertices.
- Michiels & Cools [DCG'00] describe all Minkowski summands of $\Sigma(\mathcal{A})$ including $N(\mathcal{R})$.

Algorithms

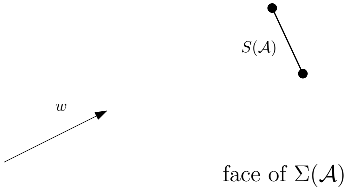
- TOPCOM [Rambau '02] computes all mixed subdivisions. Inefficient for resultant polytope computation.
- Tropical geometry [Sturmfels-Yu '08] leads to algorithms for the polytope of (specialized) resultant based on the `GFan` library [Jensen-Yu '11].

Vertex$\Pi(\mathcal{A}, w)$

Input: $\mathcal{A} \subset \mathbb{Z}^{2n}$, direction $w \in (\mathbb{R}^{|\mathcal{A}|})^{\times}$

Output: vertex $\in \Pi$, extremal wrt w

1. use w as a lifting to construct regular subdivision $S(\mathcal{A})$

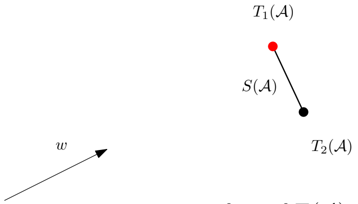


Vertex$\Pi(\mathcal{A}, w)$

Input: $\mathcal{A} \subset \mathbb{Z}^{2n}$, direction $w \in (\mathbb{R}^{|\mathcal{A}|})^{\times}$

Output: vertex $\in \Pi$, extremal wrt w

1. use w as a lifting to construct regular subdivision $S(\mathcal{A})$
2. refine $S(\mathcal{A})$ into triangulation $T(\mathcal{A})$



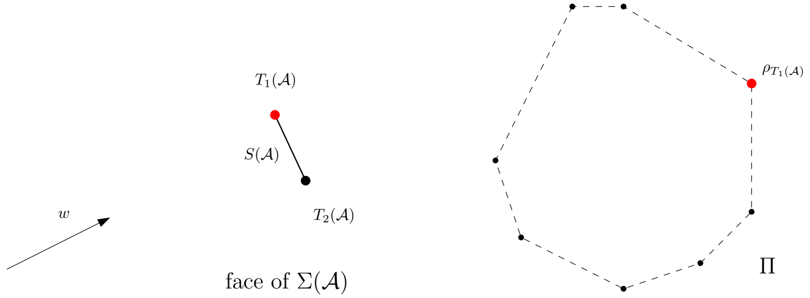
face of $\Sigma(\mathcal{A})$

Vertex$\Pi(\mathcal{A}, w)$

Input: $\mathcal{A} \subset \mathbb{Z}^{2n}$, direction $w \in (\mathbb{R}^{|\mathcal{A}|})^\times$

Output: vertex $\in \Pi$, extremal wrt w

1. use w as a lifting to construct regular subdivision $S(\mathcal{A})$
2. refine $S(\mathcal{A})$ into triangulation $T(\mathcal{A})$
3. return $\rho_{T(\mathcal{A})} \in \mathbb{N}^{|\mathcal{A}|}$

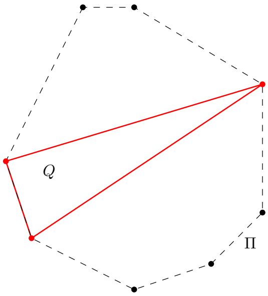


Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step



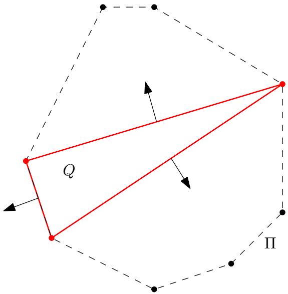
initialization:

- $Q \subset \Pi$
- $\dim(Q) = \dim(\Pi)$

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal



2 kinds of hyperplanes of Q_H :

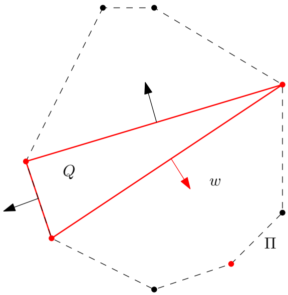
- legal if it supports a facet of Π
- illegal otherwise

Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$

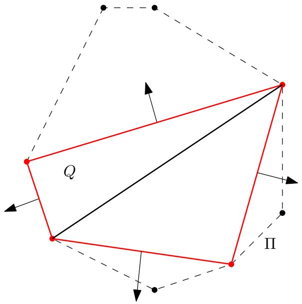


Extending an illegal facet

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ else H is legal

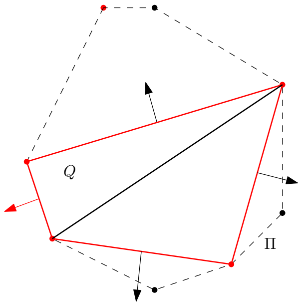


Extending an illegal facet

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ else H is legal



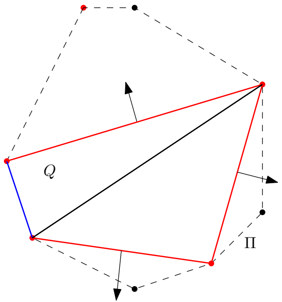
Validating a legal facet

Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ else H is legal



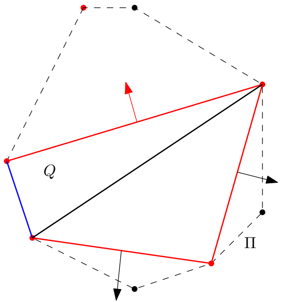
Validating a legal facet

Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ `else` H is legal

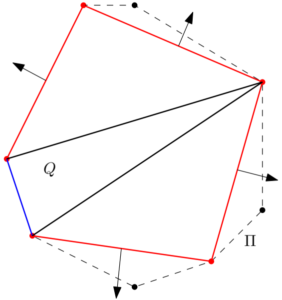


Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w `do`
    - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
    - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ `else` H is legal

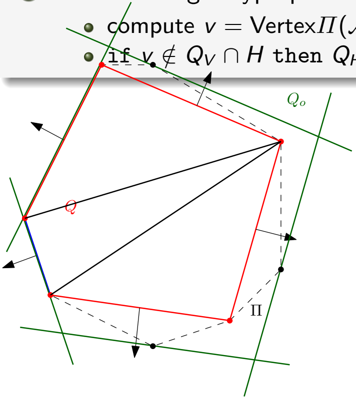


At any step, Q is an inner approximation of Π...

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ else H is legal



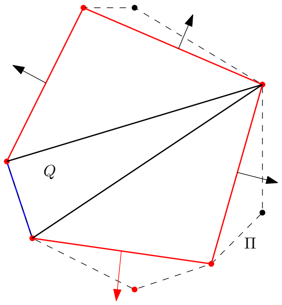
At any step, Q is an inner approximation of Π... from which we can compute an outer approximation Q_o.

Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

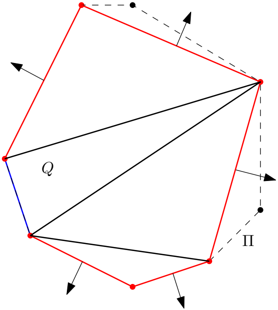
1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
    - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
    - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ else H is legal



Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

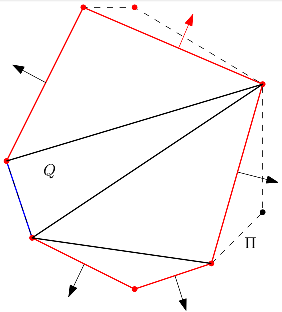
1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ else H is legal



Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
    - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
    - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ `else` H is legal

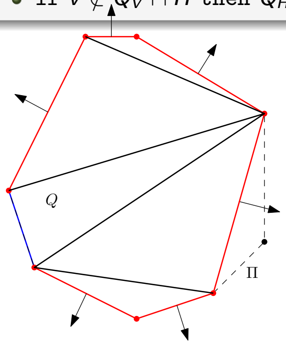


Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w `do`
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ `else` H is legal

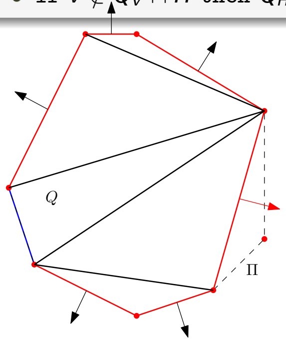


Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ `else` H is legal

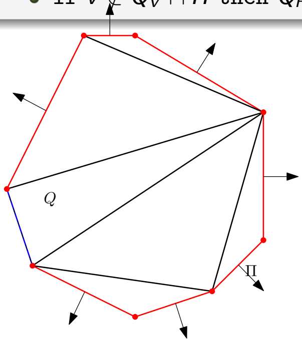


Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

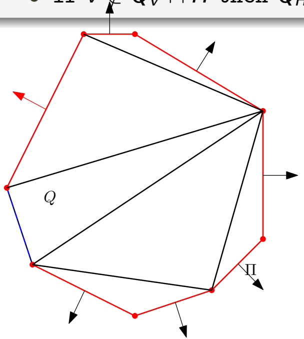
1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ else H is legal



Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

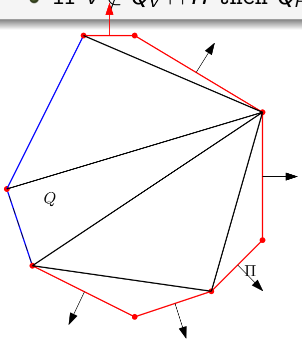
1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ else H is legal



Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

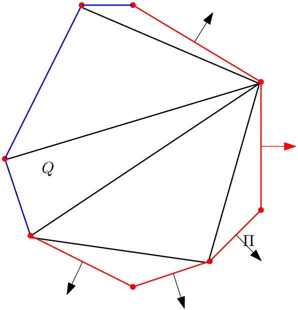
1. initialization step
2. all hyperplanes of Q_H are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w `do`
    - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
    - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ `else` H is legal



Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

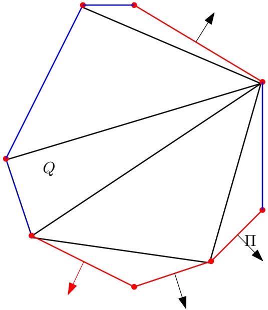
1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ else H is legal



Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

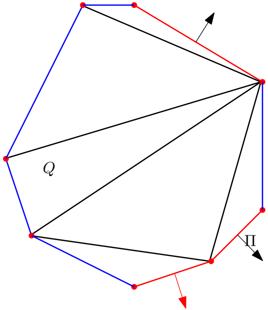
1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \mathrm{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ else H is legal



Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ `else` H is legal

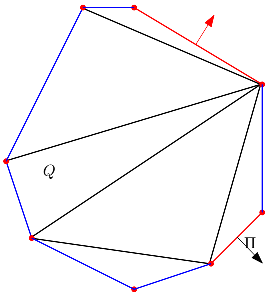


Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w `do`
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ `else` H is legal

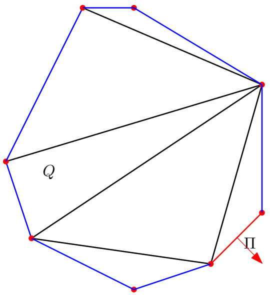


Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

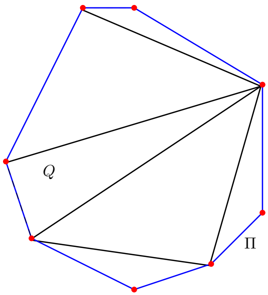
1. initialization step
2. all hyperplanes of Q_H are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ `else` H is legal



Input: $\mathcal{A}$

Output: H-rep. Q_H, V-rep. Q_V of $Q = \Pi$

1. initialization step
2. all hyperplanes of Q_H are illegal
3. while $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal w do
   - compute $v = \text{Vertex}\Pi(\mathcal{A}, w)$, $Q_V \leftarrow Q_V \cup \{v\}$
   - if $v \notin Q_V \cap H$ then $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ else H is legal



Complexity

Theorem

We compute the Vertex- and Halfspace-representations of Π, as well as a triangulation T of Π, in

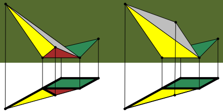
$$O^*(\, m^5\, |vtx(\Pi)| \cdot |T|^2),$$

where $m = \dim \Pi$, *and* $|T|$ *the number of full-dim faces of* T.

Elements of proof

- Most computation is done in dimension $\leq m$.
- At most $\leq \mathrm{vtx}(\Pi) + \mathrm{fct}(\Pi)$ oracle calls
- Beneath-Beyond algorithm for converting V-rep. to H-rep. (bottleneck)

ResPol Implementation



Tools

C++, CGAL, `triangulation` [Boissonnat,Devillers,Hornus], `extreme_points_d` [Gärtner]

Hashing of determinantal predicates

optimizing sequences of similar determinants

$$\widetilde{\mathcal{A}} = \begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 1 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 & w_8 & w_9 \end{vmatrix}$$

- Laplace (cofactor) expansion wrt the last row + Hash minors
- If all needed minors computed, orientation$= O(n^2)$, volume$= O(n)$

Future work

- approximate resultant polytopes ($dim(\Pi) \geq 7$)
- preliminary results:

input	m	3	3	4	4	5	5
	$\|\mathcal{A}\|$	200	490	20	30	17	20
exact	$\#$vtx(Π)	98	133	416	1296	1674	5093
	time	2.03	5.87	3.72	25.97	51.54	239.96
approx.	$\#$vtx(Q_{in})	15	11	63	121	–	–
	vol(Q_{in})/vol(Π)	0.96	0.95	0.93	0.94	–	–
	vol(Q_{out})/vol(Π)	1.02	1.03	1.04	1.03	–	–
	time	0.15	0.22	0.37	1.42	$>$ 10hr	$>$ 10hr

- approximate volume computation [Lovsz-Vempala06]

References

The code

- http://respol.sourceforge.net

The paper

- http://arxiv.org/abs/1108.5985v2 (to appear in SOCG'12)

References

The code

- `http://respol.sourceforge.net`

The paper

- `http://arxiv.org/abs/1108.5985v2` (to appear in SOCG'12)

Thank You !