# Polytopes defined by Oracles: Algorithms & Combinatorics

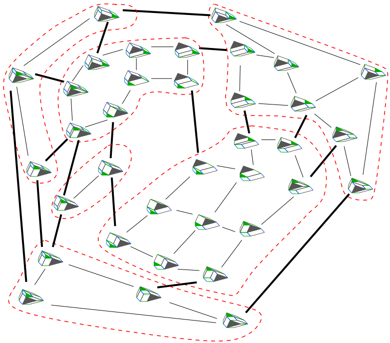
Vissarion Fisikopoulos

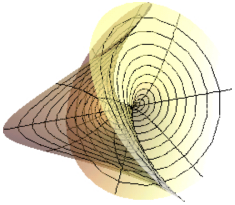
Dept. of Informatics & Telecommunications, University of Athens

University of Padova, Seminar, 12 Feb. 2014

# Main actor: resultant polytope

- Geometry: Minkowski summands of secondary polytopes, equival. classes of secondary vertices, generalize Birkhoff polytopes
- Algebra: useful to express the solvability of polynomial systems
- Applications: discriminant and resultant computation, implicitization of parametric hypersurfaces





Enneper's Minimal Surface

# Outline

Introduction: resultant polytopes

An output-sensitive algorithm for computing projections of resultant polytopes [Emiris, F, Konaxis, Peñaranda, SoCG'12]

Combinatorics of 4-d resultant polytopes [Emiris, F, Dickenstein, ISSAC'13]

Conclusion & Extensions

# Polytopes and Algebra

- Given $n+1$ polynomials on $n$ variables.

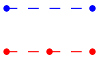
$$f_0(x) = ax^2 + b$$
$$f_1(x) = cx^2 + dx + e$$

# Polytopes and Algebra

- Given $n + 1$ polynomials on $n$ variables.
- Supports (set of exponents of monomials with non-zero coefficient) $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$.

$f_0(x) = ax^2 + b$ $\quad A_0$

$f_1(x) = cx^2 + dx + e$ $\quad A_1$



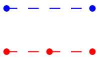
# Polytopes and Algebra

- Given $n+1$ polynomials on $n$ variables.
- Supports (set of exponents of monomials with non-zero coefficient) $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$.
- The resultant $R$ is the polynomial in the coefficients of a system of polynomials which vanishes if there exists a common root in the torus of the given polynomials.

$$f_0(x) = ax^2 + b$$
$$f_1(x) = cx^2 + dx + e$$

$A_0$

$A_1$



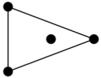
$$R(a,b,c,d,e) = ad^2b + c^2b^2 - 2caeb + a^2e^2$$

# Polytopes and Algebra

- Given $n+1$ polynomials on $n$ variables.
- Supports (set of exponents of monomials with non-zero coefficient) $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$.
- The resultant $R$ is the polynomial in the coefficients of a system of polynomials which vanishes if there exists a common root in the torus of the given polynomials.
- The resultant polytope $N(R)$, is the convex hull of the support of $R$, i.e. the Newton polytope of the resultant.

$f_0(x) = ax^2 + b$ $\qquad A_0$ 

$f_1(x) = cx^2 + dx + e$ $\qquad A_1$ 

$R(a,b,c,d,e) = ad^2b + c^2b^2 - 2caeb + a^2e^2$ $\qquad N(R)$ 

# Polytopes and Algebra

The case of linear polynomials

$$f_0(x,y) = ax + by + c$$

$A_0$

$$f_1(x,y) = dx + ey + f$$

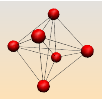
$A_1$

$$f_2(x,y) = gx + hy + i$$

$A_2$

$$R(a,b,c,d,e,f,g,h,i) = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

$N(R)$



4-dimensional Birkhoff polytope

# Polytopes and Algebra

In the general case ...

$$f_0(x,y) = axy^2 + x^4y + c$$

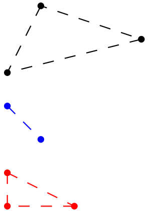
$$f_1(x,y) = dx + ey$$

$$f_2(x,y) = gx^2 + hy + i$$

$A_0$

$A_1$

$A_2$



Q1: How can we compute $N(R)$ ?

Q2: How $N(R)$ looks like ?

# Outline

Introduction: resultant polytopes

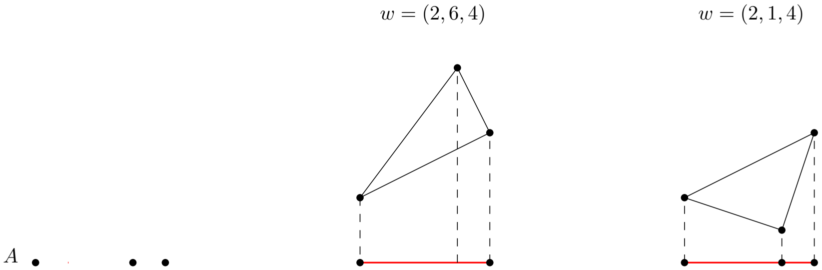
An output-sensitive algorithm for computing projections of resultant polytopes [Emiris, F, Konaxis, Peñaranda, SoCG'12]

Combinatorics of 4-d resultant polytopes [Emiris, F, Dickenstein, ISSAC'13]

Conclusion & Extensions

# Regularity and subdivisions

Regular subdivisions of $A \subset \mathbb{R}^d$ are obtained by projecting the lower (or upper) hull of $A$ lifted to $\mathbb{R}^{d+1}$ via a lifting function $w \in (\mathbb{R}^{|A|})^\times$.

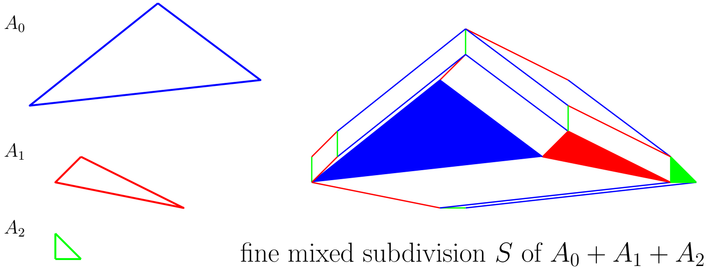


# Resultant polytope vertices and mixed subdivisions

A subdivision $S$ of $\mathcal{A} = A_0 + A_1 + \cdots + A_n$ is

- mixed when each cell is Minkowski sum of convex hulls of point subsets in $A_i$'s,
- fine when each cell has dimension equal to the sum of its summands dimensions.

## Example




fine mixed subdivision $S$ of $A_0 + A_1 + A_2$

# Resultant polytope vertices and mixed subdivisions

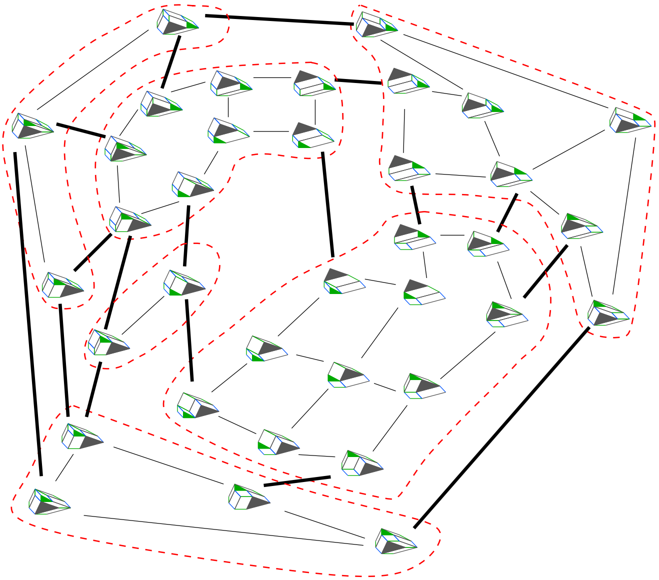
A subdivision $S$ of $\mathcal{A} = A_0 + A_1 + \cdots + A_n$ is

- mixed when each cell is Minkowski sum of convex hulls of point subsets in $A_i$'s,
- fine when each cell has dimension equal to the sum of its summands dimensions.

## Theorem [GKZ '94]

- regular fine mixed subdivisions of $\mathcal{A}$ are in one-to-one relation with the vertices of the secondary polytope $\Sigma(\mathcal{A})$
- $\exists$ a many-to-one relation between regular fine mixed subdivisions of $\mathcal{A}$ and $N(R)$ vertices

# Don't lose sight of the forest for the trees . . .



# Existing work

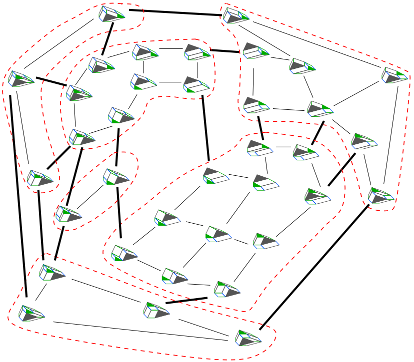
- Theory of resultants, secondary polytopes, Cayley trick [GKZ '94]
- TOPCOM [Rambau '02] computes all vertices of secondary polytope.
- [Michiels & Verschelde DCG'99] coarse equivalence classes of secondary polytope vertices.
- [Michiels & Cools DCG'00] decomposition of $\Sigma(\mathcal{A})$ in Minkowski summands, including $\mathrm{N}(\mathcal{R})$.
- Tropical geometry [Sturmfels-Yu '08]: algorithms for resultant polytope (`GFan` library) [Jensen-Yu '11] and discriminant polytope (`TropLi` software) [Rincòn '12].

# The idea of the algorithm

Input: $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$ (recall: $\mathcal{A} = A_0 + A_1 + \cdots + A_n \subset \mathbb{Z}^n$)
Simplistic method:

- compute the secondary polytope $\Sigma(\mathcal{A})$
- many-to-one relation between vertices of $\Sigma(\mathcal{A})$ and $N(R)$ vertices

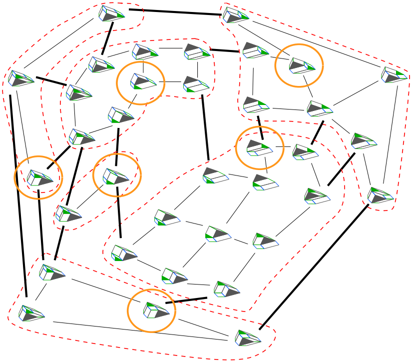
Cannot enumerate 1 representative per class by walking on secondary edges



# The idea of the algorithm

Input: $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$ (recall: $\mathcal{A} = A_0 + A_1 + \cdots + A_n \subset \mathbb{Z}^n$)
New Algorithm:

- Vertex oracle: given a direction vector compute a vertex of $N(R)$
- Output sensitive: computes only one r.f.m. subdivision of $\mathcal{A}$ per $N(R)$ vertex + one per $N(R)$ facet
- Computes projections of $N(R)$ or $\Sigma(\mathcal{A})$

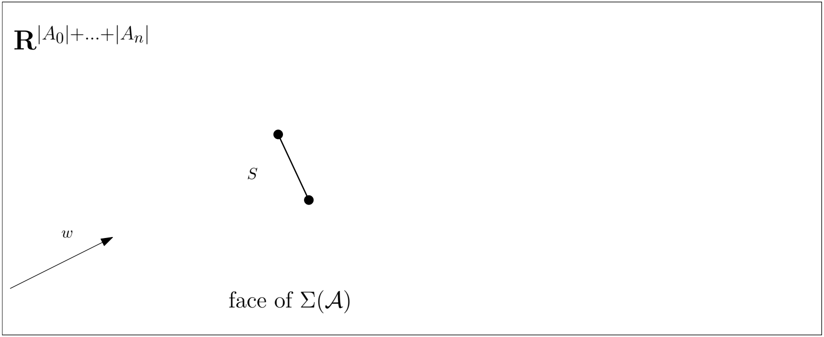


# The Vertex (Optimization) Oracle

**Input:** $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$, direction $w \in (\mathbb{R}^{|A_0|+\cdots+|A_n|})^\times$

**Output:** vertex $\in N(R)$, extremal wrt $w$

1. use $w$ as a lifting to construct r. m. subdivision $S$ of $\mathcal{A}$

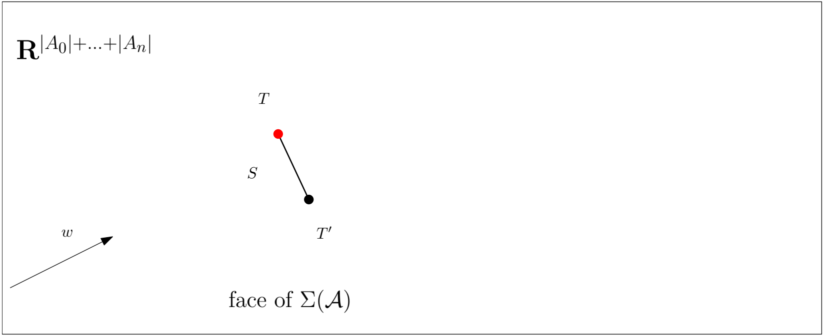


# The Vertex (Optimization) Oracle

**Input:** $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$, direction $w \in (\mathbb{R}^{|A_0|+\cdots+|A_n|})^\times$

**Output:** vertex $\in N(R)$, extremal wrt $w$

1. use $w$ as a lifting to construct r. m. subdivision $S$ of $\mathcal{A}$
2. refine $S$ into fine r.m. subdivision T of $\mathcal{A}$

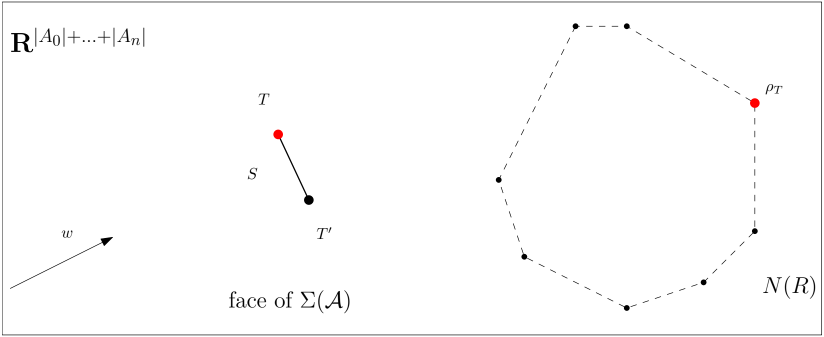


# The Vertex (Optimization) Oracle

**Input:** $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$, direction $w \in (\mathbb{R}^{|A_0|+\cdots+|A_n|})^\times$

**Output:** vertex $\in N(R)$, extremal wrt $w$

1. use $w$ as a lifting to construct r. m. subdivision $S$ of $\mathcal{A}$
2. refine $S$ into fine r.m. subdivision $T$ of $\mathcal{A}$
3. return $\rho_T \in \mathbb{N}^{|A_0|+\cdots+|A_n|}$



# The Vertex (Optimization) Oracle

Input: $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$, direction $w \in (\mathbb{R}^{|A_0|+\cdots+|A_n|})^\times$

Output: vertex $\in N(R)$, extremal wrt $w$

1. use $w$ as a lifting to construct r. m. subdivision $S$ of $\mathcal{A}$
2. refine $S$ into fine r.m. subdivision $T$ of $\mathcal{A}$
3. return $\rho_T \in \mathbb{N}^{|A_0|+\cdots+|A_n|}$

## Lemma

*Oracle's output is*

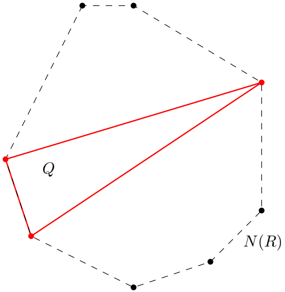
- *always a vertex of the target polytope,*
- *extremal wrt $w$.*

# Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

1. initialization step



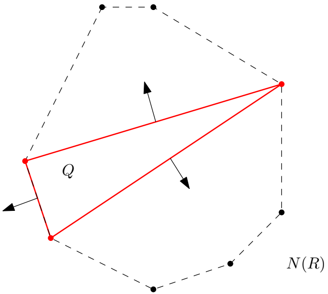

initialization:

- $Q \subset N(R)$
- dim($Q$)=dim($N(R)$)

# Incremental Algorithm

Input: $\mathcal{A}$
Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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




2 kinds of hyperplanes of $Q_H$ :

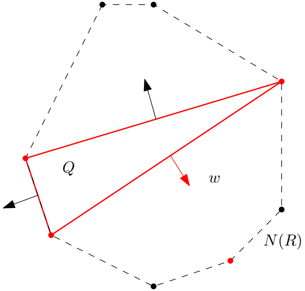
- legal if it supports facet $\subset N(R)$
- illegal otherwise

# Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$



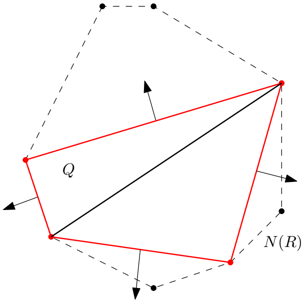

Extending an illegal facet

# Incremental Algorithm

**Input:** $\mathcal{A}$

**Output:** H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal



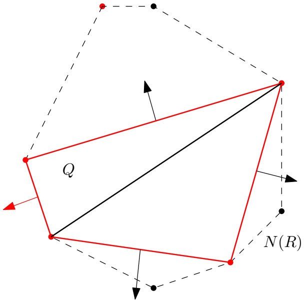

Extending an illegal facet

# Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

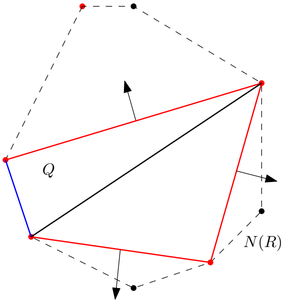



Validating a legal facet

# Incremental Algorithm

Input: $\mathcal{A}$
Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal



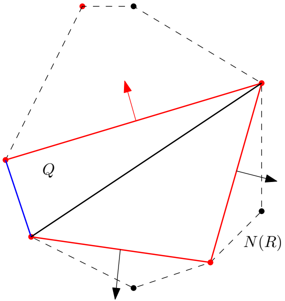

Validating a legal facet

# Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

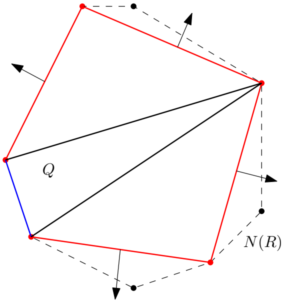


# Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

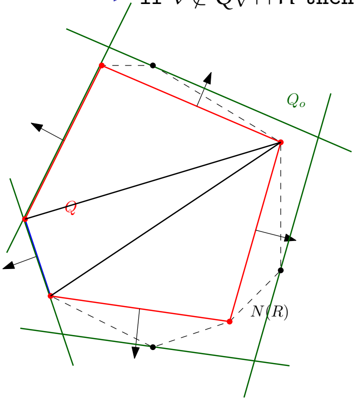



At any step, $Q$ is an inner approximation ...

# Incremental Algorithm

Input: $\mathcal{A}$
Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \text{CH}(Q_V \cup \{v\})$ `else` $H$ is legal



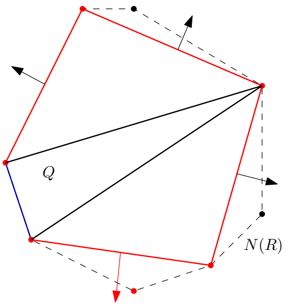

At any step, $Q$ is an inner approximation ... from which we can compute an outer approximation $Q_o$.

# Incremental Algorithm

**Input:** $\mathcal{A}$

**Output:** H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

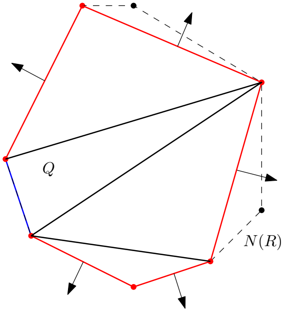


# Incremental Algorithm

**Input:** $\mathcal{A}$

**Output:** H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

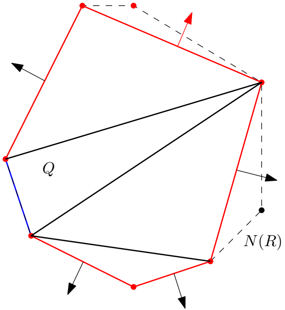


# Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

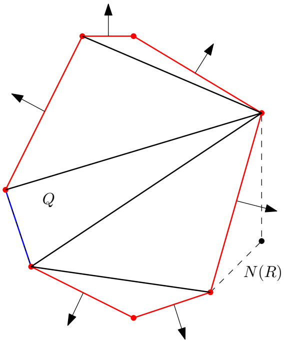


# Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

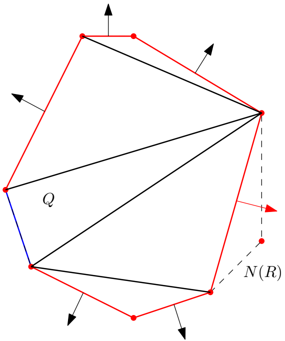


# Incremental Algorithm

**Input:** $\mathcal{A}$

**Output:** H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

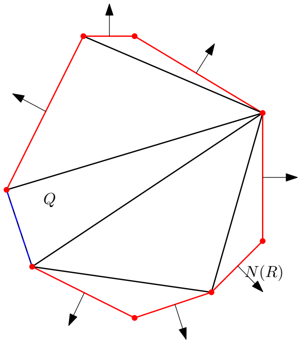


# Incremental Algorithm

**Input:** $\mathcal{A}$

**Output:** H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

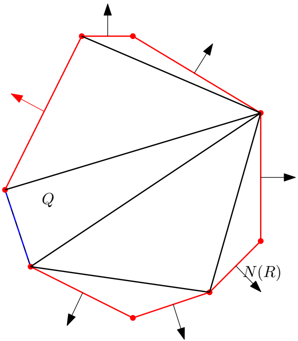


# Incremental Algorithm

Input: $\mathcal{A}$

Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

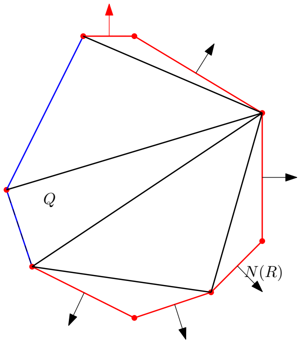


# Incremental Algorithm

**Input:** $\mathcal{A}$

**Output:** H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

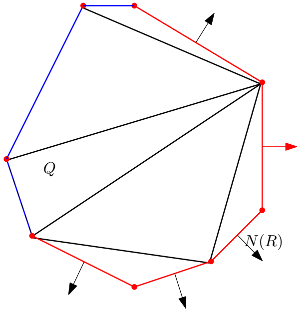
1. initialization step
2. all hyperplanes of $Q_H$ are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal $w$ `do`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$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



# Incremental Algorithm

**Input:** $\mathcal{A}$
**Output:** H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

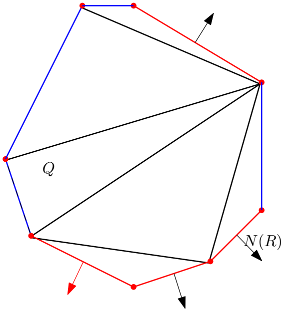


# Incremental Algorithm

**Input:** $\mathcal{A}$

**Output:** H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$ `else` $H$ is legal

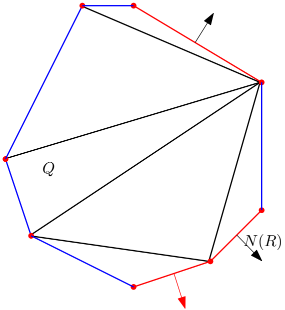


# Incremental Algorithm

**Input:** $\mathcal{A}$

**Output:** H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

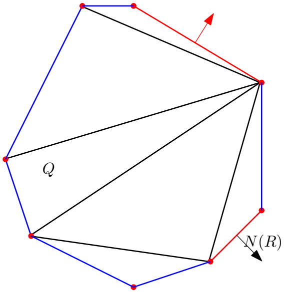
1. initialization step
2. all hyperplanes of $Q_H$ are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal $w$ `do`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$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



# Incremental Algorithm

Input: $\mathcal{A}$
Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

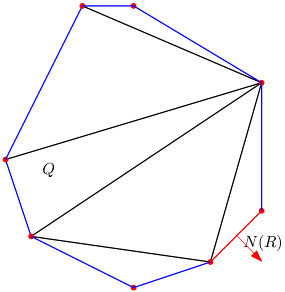
1. initialization step
2. all hyperplanes of $Q_H$ are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal $w$ `do`
    - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
    - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$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



# Incremental Algorithm

**Input:** $\mathcal{A}$
**Output:** H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

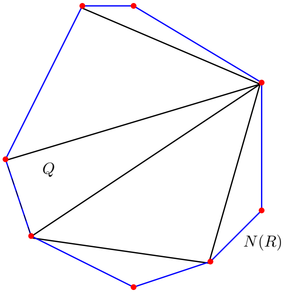
1. initialization step
2. all hyperplanes of $Q_H$ are illegal
3. `while` $\exists$ illegal hyperplane $H \subset Q_H$ with outer normal $w$ `do`
    - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
    - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow \mathrm{CH}(Q_V \cup \{v\})$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



# Incremental Algorithm

Input: $\mathcal{A}$
Output: H-rep. $Q_H$, V-rep. $Q_V$ of $Q = N(R)$

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`
   - call oracle for $w$ and compute $v$, $Q_V \leftarrow Q_V \cup \{v\}$
   - `if` $v \notin Q_V \cap H$ `then` $Q_H \leftarrow CH(Q_V \cup \{v\})$ `else` $H$ is legal



# Complexity

### Theorem

*We compute the Vertex- and Halfspace-representations of* $N(R)$*, as well as a triangulation* $T$ *of* $N(R)$*, in*

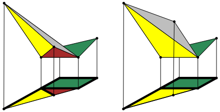
$$O^*(\, m^5\, |vtx(N(R))| \cdot |T|^2),$$

*where* $m = \dim N(R)$*, and* $|T|$ *the number of full-dim faces of* $T$*.*

### Elements of proof

- Computation is done in dimension $m = |A_0| + \cdots + |A_n| - 2n + 1$, $N(R) \subset \mathbb{R}^{|A_0|+\cdots+|A_n|}$.
- At most $\leq$ vtx$(N(R))$ + fct$(N(R))$ oracle calls.
- Beneath-and-Beyond algorithm for converting V-rep. to H-rep.

# ResPol package



- C++
- Towards high-dimensional CGAL
- Propose *hashing of determinantal predicates* scheme: optimizing sequences of similar determinants (x100 speed-up)
- Computes 5-, 6- and 7-dimensional polytopes with 35K, 23K and 500 vertices, respectively, within 2hrs
- Computes polytopes of many important surface equations encountered in geometric modeling in $<$ 1sec, whereas the corresponding secondary polytopes are intractable
- `http://sourceforge.net/projects/respol`

# Outline

Introduction: resultant polytopes

An output-sensitive algorithm for computing projections of resultant polytopes [Emiris, F, Konaxis, Peñaranda, SoCG'12]

Combinatorics of 4-d resultant polytopes [Emiris, F, Dickenstein, ISSAC'13]

Conclusion & Extensions

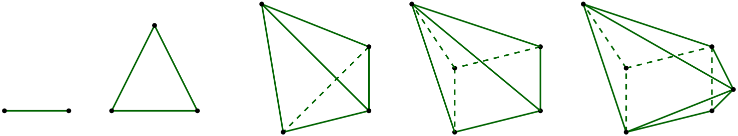
# Existing work

- [GKZ'90] Univariate case / general dimensional $\mathbb{N}(\mathbb{R})$

# Existing work

- [GKZ'90] Univariate case / general dimensional $N(R)$

- [Sturmfels'94] Multivariate case / up to $3$ dimensional $N(R)$



# One step beyond... 4-dimensional N(R)

- Polytope $P \subseteq \mathbb{R}^4$; f-vector is the vector of its face cardinalities.

# One step beyond... 4-dimensional N(R)

- Polytope $P \subseteq \mathbb{R}^4$; f-vector is the vector of its face cardinalities.
- Call vertices, edges, ridges, facets, the $0{,}1{,}2{,}3$-$d$, resp., faces of $P$.

# One step beyond... 4-dimensional N(R)

- Polytope $P \subseteq \mathbb{R}^4$; f-vector is the vector of its face cardinalities.
- Call vertices, edges, ridges, facets, the $0{,}1{,}2{,}3$-$d$, resp., faces of $P$.
- f-vectors of 4-dimensional N(R) (computed with `ResPol`)

| | |
|---|---|
| (5, 10, 10, 5) | (18, 53, 53, 18) |
| (6, 15, 18, 9) | (18, 54, 54, 18) |
| (8, 20, 21, 9) | (19, 54, 52, 17) |
| (9, 22, 21, 8) | (19, 55, 51, 15) |
| . | (19, 55, 52, 16) |
| . | (19, 55, 54, 18) |
| . | (19, 56, 54, 17) |
| (17, 49, 48, 16) | (19, 56, 56, 19) |
| (17, 49, 49, 17) | (19, 57, 57, 19) |
| (17, 50, 50, 17) | (20, 58, 54, 16) |
| (18, 51, 48, 15) | (20, 59, 57, 18) |
| (18, 51, 49, 16) | (20, 60, 60, 20) |
| (18, 52, 50, 16) | (21, 62, 60, 19) |
| (18, 52, 51, 17) | (21, 63, 63, 21) |
| (18, 53, 51, 16) | (22, 66, 66, 22) |

# Main result

### Theorem

*Given* $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$ *with* $N(R)$ *of dimension* $4$*. Then* $N(R)$ *are degenerations of the polytopes in following cases.*

- Degenarations can only decrease the number of faces.

# Main result

## Theorem

*Given* $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$ *with* $N(R)$ *of dimension* $4$. *Then* $N(R)$ *are degenerations of the polytopes in following cases.*

(i) *All* $|A_i| = 2$, *except for one with cardinality* $5$, *is a* $4$*-simplex with* $f$*-vector* $(5, 10, 10, 5)$.

- Degenarations can only decrease the number of faces.

# Main result

## Theorem

*Given* $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$ *with* $N(R)$ *of dimension* $4$*. Then* $N(R)$ *are degenerations of the polytopes in following cases.*

(i) *All* $|A_i| = 2$*, except for one with cardinality* $5$*, is a* $4$*-simplex with* $f$*-vector* $(5, 10, 10, 5)$*.*

(ii) *All* $|A_i| = 2$*, except for two with cardinalities* $3$ *and* $4$*, has* $f$*-vector* $(10, 26, 25, 9)$*.*

- Degenarations can only decrease the number of faces.

## Main result

### Theorem

*Given* $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$ *with* $N(R)$ *of dimension* $4$*. Then* $N(R)$ *are degenerations of the polytopes in following cases.*

(i) *All* $|A_i| = 2$*, except for one with cardinality* $5$*, is a* $4$*-simplex with* $f$*-vector* $(5, 10, 10, 5)$*.*

(ii) *All* $|A_i| = 2$*, except for two with cardinalities* $3$ *and* $4$*, has* $f$*-vector* $(10, 26, 25, 9)$*.*

(iii) *All* $|A_i| = 2$*, except for three with cardinality* $3$*, maximal number of ridges is* $\tilde{f}_2 = 66$ *and of facets* $\tilde{f}_3 = 22$*. Moreover,* $22 \leq \tilde{f}_0 \leq 28$*, and* $66 \leq \tilde{f}_1 \leq 72$*. The lower bounds are tight.*

- Degenarations can only decrease the number of faces.
- Focus on new case (iii), which reduces to $n = 2$ and each $|A_i| = 3$.

# Main result

## Theorem

*Given* $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$ *with* $N(R)$ *of dimension* $4$*. Then* $N(R)$ *are degenerations of the polytopes in following cases.*

(i) *All* $|A_i| = 2$*, except for one with cardinality* $5$*, is a* $4$*-simplex with* $f$*-vector* $(5, 10, 10, 5)$*.*

(ii) *All* $|A_i| = 2$*, except for two with cardinalities* $3$ *and* $4$*, has* $f$*-vector* $(10, 26, 25, 9)$*.*

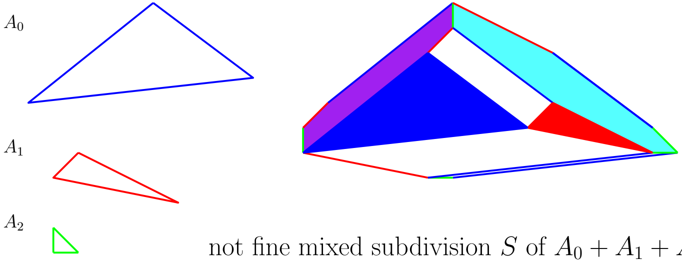
(iii) *All* $|A_i| = 2$*, except for three with cardinality* $3$*, maximal number of ridges is* $\tilde{f_2} = 66$ *and of facets* $\tilde{f_3} = 22$*. Moreover,* $22 \leq \tilde{f_0} \leq 28$*, and* $66 \leq \tilde{f_1} \leq 72$*. The lower bounds are tight.*

- Degenarations can only decrease the number of faces.
- Focus on new case (iii), which reduces to $n = 2$ and each $|A_i| = 3$.
- Previous upper bound for vertices yields 6608 [Sturmfels'94].

# Tool (1): N(R) faces and subdivisions

A subdivision $S$ of $A_0 + A_1 + \cdots + A_n$ is mixed when its cells have expressions as Minkowski sums of convex hulls of point subsets in $A_i$'s.

## Example

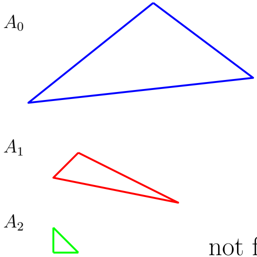


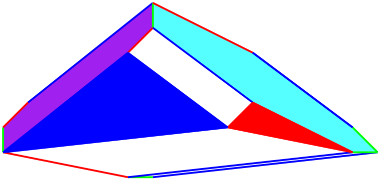

not fine mixed subdivision $S$ of $A_0 + A_1 + A_2$

# Tool (1): $N(R)$ faces and subdivisions

A subdivision $S$ of $A_0 + A_1 + \cdots + A_n$ is mixed when its cells have expressions as Minkowski sums of convex hulls of point subsets in $A_i$'s.

## Example



not fine mixed subdivision $S$ of $A_0 + A_1 + A_2$

## Proposition (Sturmfels'94)

*A regular mixed subdivision $S$ of $A_0 + A_1 + \cdots + A_n$ corresponds to a face of $N(R)$ which is the Minkowski sum of the resultant polytopes of the cells (subsystems) of $S$.*

# Tool (1): $\mathrm{N}(\mathrm{R})$ faces and subdivisions

## Example

- white, blue, red cells $\longrightarrow$ $\mathrm{N}(\mathrm{R})$ vertex

# Tool (1): $\mathcal{N}(R)$ faces and subdivisions

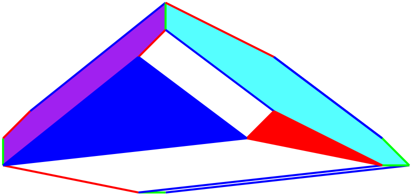
## Example

- white, blue, red cells $\rightarrow \mathcal{N}(R)$ vertex
- purple cell $\rightarrow \mathcal{N}(R)$ segment

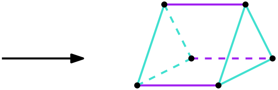
# Tool (1): N(R) faces and subdivisions

## Example

- white, blue, red cells $\longrightarrow$ N(R) vertex
- purple cell $\longrightarrow$ N(R) segment
- turquoise cell $\longrightarrow$ N(R) triangle



subd. $S$ of $A_0 + A_1 + A_2$



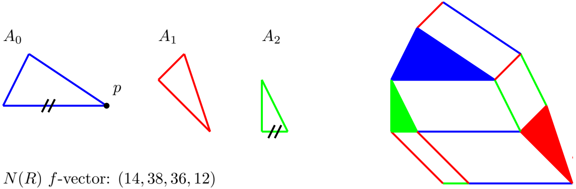
Mink. sum of $N(R)$ triangle and $N(R)$ segment

# Tool (2): Input genericity

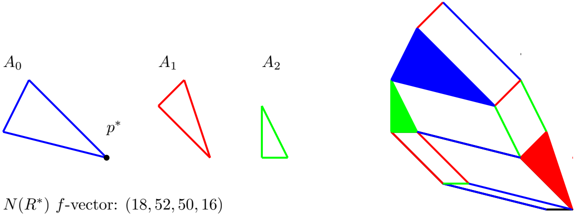
## Proposition

*Input genericity maximizes the number of resultant polytope faces.*

## Proof idea




$N(R)$ $f$-vector: $(14, 38, 36, 12)$



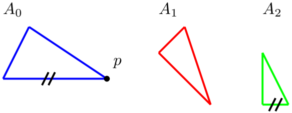

$N(R^*)$ $f$-vector: $(18, 52, 50, 16)$

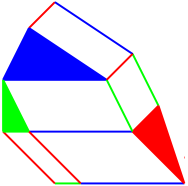
# Tool (2): Input genericity

## Proposition

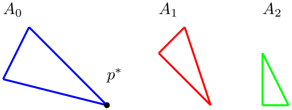
*Input genericity maximizes the number of resultant polytope faces.*

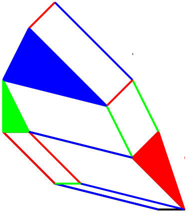
## Proof idea



$N(R)$ $f$-vector: $(14, 38, 36, 12)$



$N(R^*)$ $f$-vector: $(18, 52, 50, 16)$

$\rightarrow$ For upper bounds on the number of $\mathrm{N}(\mathrm{R})$ faces consider generic inputs, i.e. no parallel edges.

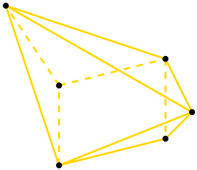
# Facets of 4-d resultant polytopes

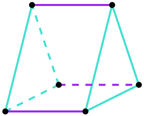
## Lemma

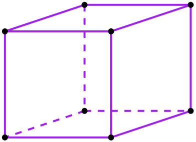
*All the possible types of* $\mathrm{N}(\mathrm{R})$ *facets are*

- *resultant facet: 3-d N(R)*
- *prism facet: 2-d N(R) (triangle)* $+$ *1-d N(R)*
- *cube facet: 1-d N(R)* $+$ *1-d N(R)* $+$ *1-d N(R)*

$3D$





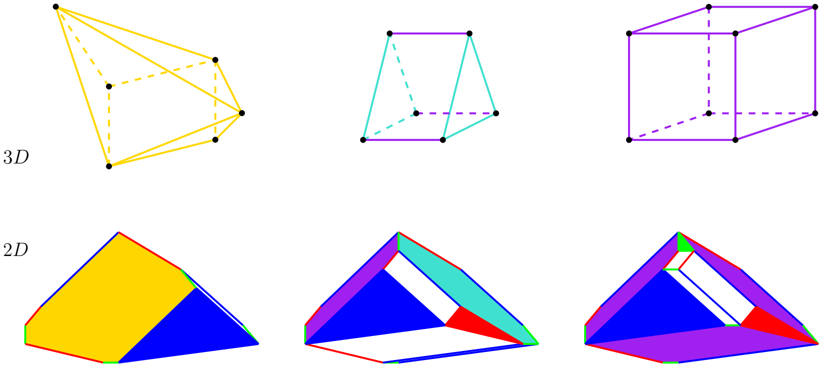


# Facets of 4-d resultant polytopes

## Lemma

*All the possible types of* $\mathrm{N}(\mathrm{R})$ *facets are*

- *resultant facet: 3-d N(R)*
- *prism facet: 2-d N(R) (triangle)* $+$ *1-d N(R)*
- *cube facet: 1-d N(R)* $+$ *1-d N(R)* $+$ *1-d N(R)*



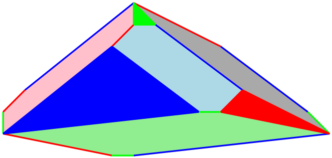
# Counting facets

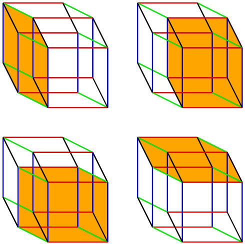
## Lemma

*There can be at most $9, 9, 4$ resultant, prism, cube facets, resp., and this is tight.*

## Proof idea

- Unique subdivision that corresponds to $4$ cube facets





# Faces of 4-d resultant polytopes

## Lemma

*The maximal number of ridges of* $N(R)$ *is* $\tilde{f_2} = 66$. *Moreover,* $\tilde{f_1} = \tilde{f_0} + 44$, $22 \leq \tilde{f_0} \leq 28$, *and* $66 \leq \tilde{f_1} \leq 72$. *The lower bounds are tight.*

## Elements of proof

- [Kalai87]

$$f_1 + \sum_{i \geq 4} (i-3) f_2^i \geq d f_0 - \binom{d+1}{2},$$

where $f_2^i$ is the number of 2-faces which are $i$-gons.

# Outline

Introduction: resultant polytopes

An output-sensitive algorithm for computing projections of resultant polytopes [Emiris, F, Konaxis, Peñaranda, SoCG'12]

Combinatorics of 4-d resultant polytopes [Emiris, F, Dickenstein, ISSAC'13]

Conclusion & Extensions

# Q3: Efficient/practical computation in high dimensions?

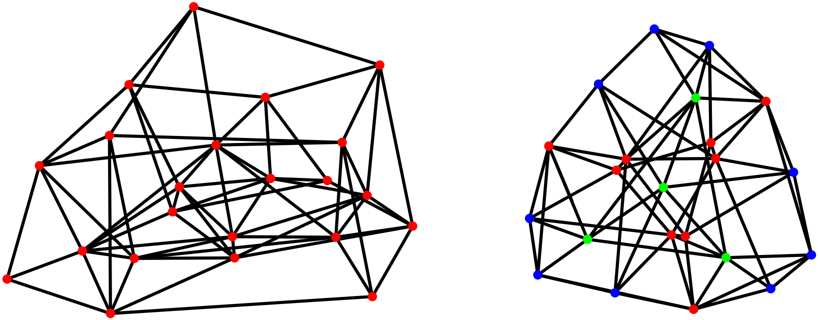
1. Edge skeleton computation [Emiris, F, Gärtner '14]

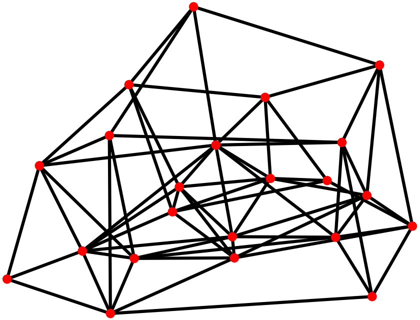
   - Input: oracle for polytope $P$ + edge directions of $P$
   - Output: edge skeleton of $P$
   - Oracle polynomial time in (input + output) size

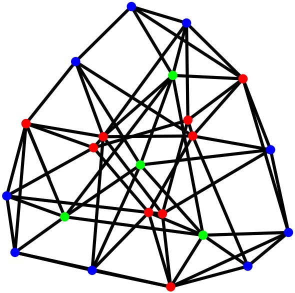
# Q3: Efficient/practical computation in high dimensions?

2. Randomized volume approximation [Emiris, F '14]

- Input: H-representation of polytope $P$
- Output: $\epsilon$-approximation of volume of $P$ for fixed error $\epsilon$
- Compute with $<$1% error the volume of several polytopes up to dimension 100 in $<$1hr whereas exact software can compute up to dimension 15
- Compute the volume of Birkhoff polytopes $B_{11}, \ldots, B_{15}$ in few hrs whereas exact methods have only computed that of $B_{10}$ by specialized parallel software in a sequential time of years







Thank you!