

Computing Monomial Bases of Highest-Weight and Demazure Modules

Xin Fang^{1,2}, Ghislain Fourier^{1,2}, Lars Götting^{1,2}, Janika Peters², Ben Wilop¹

RWTH Aachen University

¹ arXiv:2403.15018 ² ongoing work / paper in preparation



September 8, 2026

Highest weight modules

- $\mathfrak{g}$ a simple, finite-dimensional Lie algebra over $\mathbb{C}$, with triangular decomposition

$$\mathfrak{g} = \mathfrak{n}^+ \oplus \mathfrak{h} \oplus \mathfrak{n}^-$$

Highest weight modules

- $\mathfrak{g}$ a simple, finite-dimensional Lie algebra over $\mathbb{C}$, with triangular decomposition

$$\mathfrak{g} = \mathfrak{n}^+ \oplus \mathfrak{h} \oplus \mathfrak{n}^-$$

- λ a dominant integral weight

Highest weight modules

- $\mathfrak{g}$ a simple, finite-dimensional Lie algebra over $\mathbb{C}$, with triangular decomposition

$$\mathfrak{g} = \mathfrak{n}^+ \oplus \mathfrak{h} \oplus \mathfrak{n}^-$$

- λ a dominant integral weight
- $V(\lambda)$ the simple module of highest weight λ , generated by a highest weight vector v_λ :

$$V(\lambda) = \mathcal{U}(\mathfrak{n}^-).v_\lambda$$

Highest weight modules

- $\mathfrak{g}$ a simple, finite-dimensional Lie algebra over $\mathbb{C}$, with triangular decomposition

$$\mathfrak{g} = \mathfrak{n}^+ \oplus \mathfrak{h} \oplus \mathfrak{n}^-$$

- λ a dominant integral weight
- $V(\lambda)$ the simple module of highest weight λ , generated by a highest weight vector v_λ :

$$V(\lambda) = \mathcal{U}(\mathfrak{n}^-).v_\lambda$$

Classical results (Weyl, Freudenthal)

$\dim V(\lambda)$ and the character $\text{ch } V(\lambda)$ are given by explicit formulas.

Highest weight modules

- $\mathfrak{g}$ a simple, finite-dimensional Lie algebra over $\mathbb{C}$, with triangular decomposition

$$\mathfrak{g} = \mathfrak{n}^+ \oplus \mathfrak{h} \oplus \mathfrak{n}^-$$

- λ a dominant integral weight
- $V(\lambda)$ the simple module of highest weight λ , generated by a highest weight vector v_λ :

$$V(\lambda) = \mathcal{U}(\mathfrak{n}^-) \cdot v_\lambda$$

Classical results (Weyl, Freudenthal)

$\dim V(\lambda)$ and the character $\text{ch } V(\lambda)$ are given by explicit formulas.

But: neither gives an **explicit basis** of $V(\lambda)$.

Why monomial bases?

Finding an explicit basis of $V(\lambda)$ is one of the classical questions of representation theory:

- Young tableaux (Young; Hodge), Gelfand–Tsetlin bases (Gelfand, Tsetlin)
- standard monomial bases (Lakshmibai, Seshadri, 1986; Littelmann, 1998; Chirivì, Fang, Littelmann, 2023)
- canonical and global crystal bases (Lusztig, 1990; Kashiwara, 1993)
- MV bases (Mirković, Vilonen, 1988), FFLV bases (Feigin, Fourier, Littelmann, 2011), theta bases (Gross, Hacking, Keel, Kontsevich, 2018), ...

Why monomial bases?

Finding an explicit basis of $V(\lambda)$ is one of the classical questions of representation theory:

- Young tableaux (Young; Hodge), Gelfand–Tsetlin bases (Gelfand, Tsetlin)
- standard monomial bases (Lakshmibai, Seshadri, 1986; Littelmann, 1998; Chirivì, Fang, Littelmann, 2023)
- canonical and global crystal bases (Lusztig, 1990; Kashiwara, 1993)
- MV bases (Mirković, Vilonen, 1988), FFLV bases (Feigin, Fourier, Littelmann, 2011), theta bases (Gross, Hacking, Keel, Kontsevich, 2018), ...

These come from different geometrizations of $\mathcal{U}(\mathfrak{g})$ and its representations
 $\rightsquigarrow$ their computation is usually hard.

Why monomial bases?

Finding an explicit basis of $V(\lambda)$ is one of the classical questions of representation theory:

- Young tableaux (Young; Hodge), Gelfand–Tsetlin bases (Gelfand, Tsetlin)
- standard monomial bases (Lakshmibai, Seshadri, 1986; Littelmann, 1998; Chirivì, Fang, Littelmann, 2023)
- canonical and global crystal bases (Lusztig, 1990; Kashiwara, 1993)
- MV bases (Mirković, Vilonen, 1988), FFLV bases (Feigin, Fourier, Littelmann, 2011), theta bases (Gross, Hacking, Keel, Kontsevich, 2018), ...

These come from different geometrizations of $\mathcal{U}(\mathfrak{g})$ and its representations
 $\rightsquigarrow$ their computation is usually hard.

Want: a uniform construction of monomial bases that is computationally accessible

Essential exponents

Fix a **birational sequence** $(\beta_1, \dots, \beta_N)$ of positive roots, i.e. the multiplication map $U_{-\beta_1} \times \dots \times U_{-\beta_N} \rightarrow U^-$ is birational ($U_{-\beta} := \exp(\mathfrak{g}_{-\beta})$, $U^- := \exp(\mathfrak{n}^-)$)
e.g. all positive roots in any order, or simple roots along a reduced expression of w_0 .

Essential exponents

Fix a **birational sequence** $(\beta_1, \dots, \beta_N)$ of positive roots, i.e. the multiplication map $U_{-\beta_1} \times \dots \times U_{-\beta_N} \rightarrow U^-$ is birational ($U_{-\beta} := \exp(\mathfrak{g}_{-\beta})$, $U^- := \exp(\mathfrak{n}^-)$)
e.g. all positive roots in any order, or simple roots along a reduced expression of w_0 .
For $a = (a_1, \dots, a_N) \in \mathbb{Z}_{\geq 0}^N$ write $f^a := f_{\beta_1}^{a_1} \dots f_{\beta_N}^{a_N} \in \mathcal{U}(\mathfrak{n}^-)$.

Essential exponents

Fix a **birational sequence** $(\beta_1, \dots, \beta_N)$ of positive roots, i.e. the multiplication map $U_{-\beta_1} \times \dots \times U_{-\beta_N} \rightarrow U^-$ is birational ($U_{-\beta} := \exp(\mathfrak{g}_{-\beta})$, $U^- := \exp(\mathfrak{n}^-)$)
e.g. all positive roots in any order, or simple roots along a reduced expression of w_0 .
For $a = (a_1, \dots, a_N) \in \mathbb{Z}_{\geq 0}^N$ write $f^a := f_{\beta_1}^{a_1} \dots f_{\beta_N}^{a_N} \in \mathcal{U}(\mathfrak{n}^-)$.

Goal

Find $S(\lambda) \subseteq \mathbb{Z}_{\geq 0}^N$ such that $\{f^a \cdot v_\lambda \mid a \in S(\lambda)\}$ is a basis of $V(\lambda)$.

Essential exponents

Fix a **birational sequence** $(\beta_1, \dots, \beta_N)$ of positive roots, i.e. the multiplication map $U_{-\beta_1} \times \dots \times U_{-\beta_N} \rightarrow U^-$ is birational ($U_{-\beta} := \exp(\mathfrak{g}_{-\beta})$, $U^- := \exp(\mathfrak{n}^-)$)
e.g. all positive roots in any order, or simple roots along a reduced expression of w_0 .
For $a = (a_1, \dots, a_N) \in \mathbb{Z}_{\geq 0}^N$ write $f^a := f_{\beta_1}^{a_1} \dots f_{\beta_N}^{a_N} \in \mathcal{U}(\mathfrak{n}^-)$.

Goal

Find $S(\lambda) \subseteq \mathbb{Z}_{\geq 0}^N$ such that $\{f^a \cdot v_\lambda \mid a \in S(\lambda)\}$ is a basis of $V(\lambda)$.

Fix a **monomial order** $\prec$ on $\mathbb{Z}_{\geq 0}^N$.

Essential exponents

Fix a **birational sequence** $(\beta_1, \dots, \beta_N)$ of positive roots, i.e. the multiplication map $U_{-\beta_1} \times \dots \times U_{-\beta_N} \rightarrow U^-$ is birational ($U_{-\beta} := \exp(\mathfrak{g}_{-\beta})$, $U^- := \exp(\mathfrak{n}^-)$)
e.g. all positive roots in any order, or simple roots along a reduced expression of w_0 .
For $a = (a_1, \dots, a_N) \in \mathbb{Z}_{\geq 0}^N$ write $f^a := f_{\beta_1}^{a_1} \dots f_{\beta_N}^{a_N} \in \mathcal{U}(\mathfrak{n}^-)$.

Goal

Find $S(\lambda) \subseteq \mathbb{Z}_{\geq 0}^N$ such that $\{f^a.v_\lambda \mid a \in S(\lambda)\}$ is a basis of $V(\lambda)$.

Fix a **monomial order** $\prec$ on $\mathbb{Z}_{\geq 0}^N$.

Canonical choice: essential exponents (Fang, Fourier, Littelmann, 2017)

$a \in \mathbb{Z}_{\geq 0}^N$ is **essential** if $f^a.v_\lambda \notin \langle f^b.v_\lambda \mid b \prec a \rangle_{\mathbb{C}}$.

Essential exponents

Fix a **birational sequence** $(\beta_1, \dots, \beta_N)$ of positive roots, i.e. the multiplication map $U_{-\beta_1} \times \dots \times U_{-\beta_N} \rightarrow U^-$ is birational ($U_{-\beta} := \exp(\mathfrak{g}_{-\beta})$, $U^- := \exp(\mathfrak{n}^-)$)
e.g. all positive roots in any order, or simple roots along a reduced expression of w_0 .
For $a = (a_1, \dots, a_N) \in \mathbb{Z}_{\geq 0}^N$ write $f^a := f_{\beta_1}^{a_1} \dots f_{\beta_N}^{a_N} \in \mathcal{U}(\mathfrak{n}^-)$.

Goal

Find $S(\lambda) \subseteq \mathbb{Z}_{\geq 0}^N$ such that $\{f^a.v_\lambda \mid a \in S(\lambda)\}$ is a basis of $V(\lambda)$.

Fix a **monomial order** $\prec$ on $\mathbb{Z}_{\geq 0}^N$.

Canonical choice: essential exponents (Fang, Fourier, Littelmann, 2017)

$a \in \mathbb{Z}_{\geq 0}^N$ is **essential** if $f^a.v_\lambda \notin \langle f^b.v_\lambda \mid b \prec a \rangle_{\mathbb{C}}$.

The set $\text{es}(\lambda)$ of all essential exponents solves the problem.

A naive algorithm

The definition of essential exponents is already an algorithm:

A naive algorithm

The definition of essential exponents is already an algorithm:

1. construct $V(\lambda)$ with its $\mathcal{U}(\mathfrak{n}^-)$ -action explicitly
(e.g. inside a tensor product of fundamental modules)

A naive algorithm

The definition of essential exponents is already an algorithm:

1. construct $V(\lambda)$ with its $\mathcal{U}(\mathfrak{n}^-)$ -action explicitly
(e.g. inside a tensor product of fundamental modules)
2. iterate over all monomials f^a in increasing $\prec$ -order: (requires $\prec$ to be a global ordering)
keep a if $f^a.v_\lambda$ extends the span of the vectors kept so far

A naive algorithm

The definition of essential exponents is already an algorithm:

1. construct $V(\lambda)$ with its $\mathcal{U}(\mathfrak{n}^-)$ -action explicitly
(e.g. inside a tensor product of fundamental modules)
2. iterate over all monomials f^a in increasing $\prec$ -order: (requires $\prec$ to be a global ordering)
keep a if $f^a.v_\lambda$ extends the span of the vectors kept so far
3. stop once $\dim V(\lambda)$ vectors have been found

A naive algorithm

The definition of essential exponents is already an algorithm:

1. construct $V(\lambda)$ with its $\mathcal{U}(\mathfrak{n}^-)$ -action explicitly
(e.g. inside a tensor product of fundamental modules)
2. iterate over all monomials f^a in increasing $\prec$ -order: (requires $\prec$ to be a global ordering)
keep a if $f^a.v_\lambda$ extends the span of the vectors kept so far
3. stop once $\dim V(\lambda)$ vectors have been found

Correct, but expensive

There is a **huge number** of candidate monomials,
and for each one we need to **compute its action** in the explicit construction of $V(\lambda)$.

A naive algorithm

The definition of essential exponents is already an algorithm:

1. construct $V(\lambda)$ with its $\mathcal{U}(\mathfrak{n}^-)$ -action explicitly
(e.g. inside a tensor product of fundamental modules)
2. iterate over all monomials f^a in increasing $\prec$ -order: (requires $\prec$ to be a global ordering)
keep a if $f^a.v_\lambda$ extends the span of the vectors kept so far
3. stop once $\dim V(\lambda)$ vectors have been found

Correct, but expensive

There is a **huge number** of candidate monomials,
and for each one we need to **compute its action** in the explicit construction of $V(\lambda)$.

Another approach: Gröbner bases for $\mathcal{U}(\mathfrak{g})$ (implemented in GAP) — also expensive.

The Minkowski property

Theorem (Fang, Fourier, Littelmann, 2017)

For dominant integral weights λ and μ : $\text{es}(\lambda) + \text{es}(\mu) \subseteq \text{es}(\lambda + \mu)$ (Minkowski sum)

The Minkowski property

Theorem (Fang, Fourier, Littelmann, 2017)

For dominant integral weights λ and μ : $\text{es}(\lambda) + \text{es}(\mu) \subseteq \text{es}(\lambda + \mu)$ (Minkowski sum)

Hence for all decompositions of λ into dominant integral summands:

$$\bigcup_{\lambda=\lambda_1+\lambda_2} (\text{es}(\lambda_1) + \text{es}(\lambda_2)) \subseteq \text{es}(\lambda)$$

The Minkowski property

Theorem (Fang, Fourier, Littelmann, 2017)

For dominant integral weights λ and μ : $\text{es}(\lambda) + \text{es}(\mu) \subseteq \text{es}(\lambda + \mu)$ (Minkowski sum)

Hence for all decompositions of λ into dominant integral summands:

$$\bigcup_{\lambda=\lambda_1+\lambda_2} (\text{es}(\lambda_1) + \text{es}(\lambda_2)) \subseteq \text{es}(\lambda)$$

- **divide and conquer**: glue $\text{es}(\lambda)$ from the results for smaller weights

The Minkowski property

Theorem (Fang, Fourier, Littelmann, 2017)

For dominant integral weights λ and μ : $\text{es}(\lambda) + \text{es}(\mu) \subseteq \text{es}(\lambda + \mu)$ (Minkowski sum)

Hence for all decompositions of λ into dominant integral summands:

$$\bigcup_{\lambda=\lambda_1+\lambda_2} (\text{es}(\lambda_1) + \text{es}(\lambda_2)) \subseteq \text{es}(\lambda)$$

- **divide and conquer**: glue $\text{es}(\lambda)$ from the results for smaller weights
- $\dim V(\lambda)$ is known $\rightsquigarrow$ we can **check** whether the union is already all of $\text{es}(\lambda)$

The Minkowski property

Theorem (Fang, Fourier, Littelmann, 2017)

For dominant integral weights λ and μ : $\text{es}(\lambda) + \text{es}(\mu) \subseteq \text{es}(\lambda + \mu)$ (Minkowski sum)

Hence for all decompositions of λ into dominant integral summands:

$$\bigcup_{\lambda=\lambda_1+\lambda_2} (\text{es}(\lambda_1) + \text{es}(\lambda_2)) \subseteq \text{es}(\lambda)$$

- **divide and conquer**: glue $\text{es}(\lambda)$ from the results for smaller weights
- $\dim V(\lambda)$ is known $\rightsquigarrow$ we can **check** whether the union is already all of $\text{es}(\lambda)$
- very often it is! (best case: everything comes from the fundamental weights)

The Minkowski property

Theorem (Fang, Fourier, Littelmann, 2017)

For dominant integral weights λ and μ : $\text{es}(\lambda) + \text{es}(\mu) \subseteq \text{es}(\lambda + \mu)$ (Minkowski sum)

Hence for all decompositions of λ into dominant integral summands:

$$\bigcup_{\lambda=\lambda_1+\lambda_2} (\text{es}(\lambda_1) + \text{es}(\lambda_2)) \subseteq \text{es}(\lambda)$$

- **divide and conquer**: glue $\text{es}(\lambda)$ from the results for smaller weights
- $\dim V(\lambda)$ is known $\rightsquigarrow$ we can **check** whether the union is already all of $\text{es}(\lambda)$
- very often it is! (best case: everything comes from the fundamental weights)

Example (type A_2 , all positive roots by decreasing height, degrevlex):

$$|\text{es}(\omega_1) + \text{es}(\omega_2)| = 8 = \dim V(\omega_1 + \omega_2) \rightsquigarrow \text{es}(\omega_1 + \omega_2) \text{ for free}$$

When Minkowski is not enough

If the Minkowski union misses some monomials (or λ is a fundamental weight — the base case), the rest has to be computed directly — but smarter than before:

When Minkowski is not enough

If the Minkowski union misses some monomials (or λ is a fundamental weight — the base case), the rest has to be computed directly — but smarter than before:

Weight grading

$f^a.v_\lambda$ has weight $\mu = \lambda - \sum_i a_i \beta_i \rightsquigarrow$ the problem splits into weight spaces.

When Minkowski is not enough

If the Minkowski union misses some monomials (or λ is a fundamental weight — the base case), the rest has to be computed directly — but smarter than before:

Weight grading

$f^a \cdot v_\lambda$ has weight $\mu = \lambda - \sum_i a_i \beta_i \rightsquigarrow$ the problem splits into weight spaces.

For each weight space $V(\lambda)_\mu$ that is not yet full:

When Minkowski is not enough

If the Minkowski union misses some monomials (or λ is a fundamental weight — the base case), the rest has to be computed directly — but smarter than before:

Weight grading

$f^a \cdot v_\lambda$ has weight $\mu = \lambda - \sum_i a_i \beta_i \rightsquigarrow$ the problem splits into weight spaces.

For each weight space $V(\lambda)_\mu$ that is not yet full:

- its dimension is known (character formula)

When Minkowski is not enough

If the Minkowski union misses some monomials (or λ is a fundamental weight — the base case), the rest has to be computed directly — but smarter than before:

Weight grading

$f^a.v_\lambda$ has weight $\mu = \lambda - \sum_i a_i \beta_i \rightsquigarrow$ the problem splits into weight spaces.

For each weight space $V(\lambda)_\mu$ that is not yet full:

- its dimension is known (character formula)
- candidate monomials = lattice points of the polytope
 $\{x \in \mathbb{R}_{\geq 0}^N \mid \sum_i x_i \beta_i = \lambda - \mu\}$ (computed via an Integer Program (IP))

When Minkowski is not enough

If the Minkowski union misses some monomials (or λ is a fundamental weight — the base case), the rest has to be computed directly — but smarter than before:

Weight grading

$f^a \cdot v_\lambda$ has weight $\mu = \lambda - \sum_i a_i \beta_i \rightsquigarrow$ the problem splits into weight spaces.

For each weight space $V(\lambda)_\mu$ that is not yet full:

- its dimension is known (character formula)
- candidate monomials = lattice points of the polytope $\{x \in \mathbb{R}_{\geq 0}^N \mid \sum_i x_i \beta_i = \lambda - \mu\}$ (computed via an Integer Program (IP))
- go through them in $\prec$ -order, keep the span-extending ones, until the weight space is full

When Minkowski is not enough

If the Minkowski union misses some monomials (or λ is a fundamental weight — the base case), the rest has to be computed directly — but smarter than before:

Weight grading

$f^a.v_\lambda$ has weight $\mu = \lambda - \sum_i a_i \beta_i \rightsquigarrow$ the problem splits into weight spaces.

For each weight space $V(\lambda)_\mu$ that is not yet full:

- its dimension is known (character formula)
- candidate monomials = lattice points of the polytope
 $\{x \in \mathbb{R}_{\geq 0}^N \mid \sum_i x_i \beta_i = \lambda - \mu\}$ (computed via an Integer Program (IP))
- go through them in $\prec$ -order, keep the span-extending ones, until the weight space is full

Bonus: finite candidate sets $\rightsquigarrow$ non-global orderings are fine (e.g. neglex)

Making it fast

The full algorithm: **memoised recursion** over λ — glue by Minkowski sums, patch incomplete weight spaces directly.

Making it fast

The full algorithm: **memoised recursion** over λ — glue by Minkowski sums, patch incomplete weight spaces directly.

Implementation tricks:

Making it fast

The full algorithm: **memoised recursion** over λ — glue by Minkowski sums, patch incomplete weight spaces directly.

Implementation tricks:

- we cannot construct $V(\lambda)$ itself — instead compute inside $\bigotimes_i V(\omega_i)^{\otimes \lambda_i} \supseteq \mathcal{U}(\mathfrak{n}^-) \cdot v_\lambda \cong V(\lambda)$ (operators: sparse Kronecker sums)

Making it fast

The full algorithm: **memoised recursion** over λ — glue by Minkowski sums, patch incomplete weight spaces directly.

Implementation tricks:

- we cannot construct $V(\lambda)$ itself — instead compute inside $\bigotimes_i V(\omega_i)^{\otimes \lambda_i} \supseteq \mathcal{U}(\mathfrak{n}^-) \cdot v_\lambda \cong V(\lambda)$ (operators: sparse Kronecker sums)
- apply f^a factor by factor: skip a once an intermediate weight leaves the character

Making it fast

The full algorithm: **memoised recursion** over λ — glue by Minkowski sums, patch incomplete weight spaces directly.

Implementation tricks:

- we cannot construct $V(\lambda)$ itself — instead compute inside $\bigotimes_i V(\omega_i)^{\otimes \lambda_i} \supseteq \mathcal{U}(\mathfrak{n}^-) \cdot v_\lambda \cong V(\lambda)$ (operators: sparse Kronecker sums)
- apply f^a factor by factor: skip a once an intermediate weight leaves the character
- incremental sparse echelon forms over $\mathbb{Q}$ for the rank tests

Making it fast

The full algorithm: **memoised recursion** over λ — glue by Minkowski sums, patch incomplete weight spaces directly.

Implementation tricks:

- we cannot construct $V(\lambda)$ itself — instead compute inside $\bigotimes_i V(\omega_i)^{\otimes \lambda_i} \supseteq \mathcal{U}(\mathfrak{n}^-) \cdot v_\lambda \cong V(\lambda)$ (operators: sparse Kronecker sums)
- apply f^a factor by factor: skip a once an intermediate weight leaves the character
- incremental sparse echelon forms over $\mathbb{Q}$ for the rank tests
- find small-ish upper bounds for each exponent (to reduce the IP solution space)

Making it fast

The full algorithm: **memoised recursion** over λ — glue by Minkowski sums, patch incomplete weight spaces directly.

Implementation tricks:

- we cannot construct $V(\lambda)$ itself — instead compute inside $\bigotimes_i V(\omega_i)^{\otimes \lambda_i} \supseteq \mathcal{U}(\mathfrak{n}^-) \cdot v_\lambda \cong V(\lambda)$ (operators: sparse Kronecker sums)
- apply f^a factor by factor: skip a once an intermediate weight leaves the character
- incremental sparse echelon forms over $\mathbb{Q}$ for the rank tests
- find small-ish upper bounds for each exponent (to reduce the IP solution space)

$A_3, \lambda = (5, 5, 6), \dim V(\lambda) = 62244$ (both deglex)

GAP (Gröbner bases) ~ 81 s

OSCAR implementation ~ 0.8 s

Monomial bases in OSCAR

```
julia> basis_lie_highest_weight(:A, 2, [1, 1])  
Monomial basis of a highest weight module  
  of highest weight  $w_1 + w_2$   
  of dimension 8  
  with monomial ordering degrevlex([x1, x2, x3])  
over abstract Lie algebra of type A2 over QQ  
where the used birational sequence consists of the following roots:  
  [a_1, a_2, a_1 + a_2]
```

Many variants and customization options:

Monomial bases in OSCAR

```
julia> basis_lie_highest_weight(:A, 2, [1, 1])  
Monomial basis of a highest weight module  
  of highest weight  $w_1 + w_2$   
  of dimension 8  
  with monomial ordering degrevlex([x1, x2, x3])  
over abstract Lie algebra of type A2 over QQ  
where the used birational sequence consists of the following roots:  
  [a_1, a_2, a_1 + a_2]
```

Many variants and customization options:

- choose the **birational sequence** and the **monomial ordering** freely

Monomial bases in OSCAR

```
julia> basis_lie_highest_weight(:A, 2, [1, 1])  
Monomial basis of a highest weight module  
  of highest weight  $w_1 + w_2$   
  of dimension 8  
  with monomial ordering degrevlex([x1, x2, x3])  
over abstract Lie algebra of type A2 over QQ  
where the used birational sequence consists of the following roots:  
[a_1, a_2, a_1 + a_2]
```

Many variants and customization options:

- choose the **birational sequence** and the **monomial ordering** freely
- named variants reproduce the well-known parametrizations:
`basis_lie_highest_weight_string`, `..._lusztig`, `..._nz`, `..._ffl`

For $w \in W$, the weight space $V(\lambda)_{\lambda \cdot w}$ is one-dimensional, spanned by the extremal weight vector $v_{\lambda \cdot w}$.

Demazure modules

For $w \in W$, the weight space $V(\lambda)_{\lambda \cdot w}$ is one-dimensional, spanned by the **extremal weight vector** $v_{\lambda \cdot w}$.

Definition

The **Demazure module** $V_w(\lambda) := \mathcal{U}(\mathfrak{b}) \cdot v_{\lambda \cdot w} \subseteq V(\lambda)$ is a $\mathfrak{b}$ -submodule ($\mathfrak{b} := \mathfrak{h} \oplus \mathfrak{n}^+$).

Demazure modules

For $w \in W$, the weight space $V(\lambda)_{\lambda \cdot w}$ is one-dimensional, spanned by the **extremal weight vector** $v_{\lambda \cdot w}$.

Definition

The **Demazure module** $V_w(\lambda) := \mathcal{U}(\mathfrak{b}) \cdot v_{\lambda \cdot w} \subseteq V(\lambda)$ is a $\mathfrak{b}$ -submodule ($\mathfrak{b} := \mathfrak{h} \oplus \mathfrak{n}^+$).

- interpolation: $V_e(\lambda) = \mathbb{C}v_\lambda$ and $V_{w_0}(\lambda) = V(\lambda)$

Demazure modules

For $w \in W$, the weight space $V(\lambda)_{\lambda \cdot w}$ is one-dimensional, spanned by the **extremal weight vector** $v_{\lambda \cdot w}$.

Definition

The **Demazure module** $V_w(\lambda) := \mathcal{U}(\mathfrak{b}) \cdot v_{\lambda \cdot w} \subseteq V(\lambda)$ is a $\mathfrak{b}$ -submodule ($\mathfrak{b} := \mathfrak{h} \oplus \mathfrak{n}^+$).

- interpolation: $V_e(\lambda) = \mathbb{C}v_\lambda$ and $V_{w_0}(\lambda) = V(\lambda)$
- geometry: $V_w(\lambda) \cong H^0(X_w, \mathcal{L}_\lambda)^*$ for the **Schubert variety** $X_w = \overline{BwB/B} \subseteq G/B$

For $w \in W$, the weight space $V(\lambda)_{\lambda \cdot w}$ is one-dimensional, spanned by the **extremal weight vector** $v_{\lambda \cdot w}$.

Definition

The **Demazure module** $V_w(\lambda) := \mathcal{U}(\mathfrak{b}) \cdot v_{\lambda \cdot w} \subseteq V(\lambda)$ is a $\mathfrak{b}$ -submodule ($\mathfrak{b} := \mathfrak{h} \oplus \mathfrak{n}^+$).

- interpolation: $V_e(\lambda) = \mathbb{C}v_\lambda$ and $V_{w_0}(\lambda) = V(\lambda)$
- geometry: $V_w(\lambda) \cong H^0(X_w, \mathcal{L}_\lambda)^*$ for the **Schubert variety** $X_w = \overline{BwB/B} \subseteq G/B$
- dimension and character are given by Demazure's character formula (1974)
 $\rightsquigarrow$ in OSCAR: `demazure_character`

Demazure modules: what goes wrong

Can we compute monomial bases of $V_w(\lambda)$ with the same algorithm?

Demazure modules: what goes wrong

Can we compute monomial bases of $V_w(\lambda)$ with the same algorithm?

the algorithm for $V(\lambda)$ used

for $V_w(\lambda)$

Demazure modules: what goes wrong

Can we compute monomial bases of $V_w(\lambda)$ with the same algorithm?

the algorithm for $V(\lambda)$ used	for $V_w(\lambda)$
weight grading, candidates from polyhedra	✓ ($\mathfrak{h}$ -stable)

Demazure modules: what goes wrong

Can we compute monomial bases of $V_w(\lambda)$ with the same algorithm?

the algorithm for $V(\lambda)$ used	for $V_w(\lambda)$
weight grading, candidates from polyhedra	✓ ($\mathfrak{h}$ -stable)
dimensions of all weight spaces	✓ (Demazure character formula)

Demazure modules: what goes wrong

Can we compute monomial bases of $V_w(\lambda)$ with the same algorithm?

the algorithm for $V(\lambda)$ used	for $V_w(\lambda)$
weight grading, candidates from polyhedra	✓ ($\mathfrak{h}$ -stable)
dimensions of all weight spaces	✓ (Demazure character formula)
cyclic generator = highest weight vector	× ($v_{\lambda \cdot w}$ is extremal, not highest)

Demazure modules: what goes wrong

Can we compute monomial bases of $V_w(\lambda)$ with the same algorithm?

the algorithm for $V(\lambda)$ used	for $V_w(\lambda)$
weight grading, candidates from polyhedra	✓ ($\mathfrak{h}$ -stable)
dimensions of all weight spaces	✓ (Demazure character formula)
cyclic generator = highest weight vector	× ($v_{\lambda \cdot w}$ is extremal, not highest)
stability under the operators f_β	× (only a $\mathfrak{b}$ -module)

Demazure modules: what goes wrong

Can we compute monomial bases of $V_w(\lambda)$ with the same algorithm?

the algorithm for $V(\lambda)$ used	for $V_w(\lambda)$
weight grading, candidates from polyhedra	✓ ($\mathfrak{h}$ -stable)
dimensions of all weight spaces	✓ (Demazure character formula)
cyclic generator = highest weight vector	× ($v_{\lambda \cdot w}$ is extremal, not highest)
stability under the operators f_β	× (only a $\mathfrak{b}$ -module)
Minkowski property	?

Demazure modules: what goes wrong

Can we compute monomial bases of $V_w(\lambda)$ with the same algorithm?

the algorithm for $V(\lambda)$ used	for $V_w(\lambda)$
weight grading, candidates from polyhedra	✓ ($\mathfrak{h}$ -stable)
dimensions of all weight spaces	✓ (Demazure character formula)
cyclic generator = highest weight vector	× ($v_{\lambda \cdot w}$ is extremal, not highest)
stability under the operators f_β	× (only a $\mathfrak{b}$ -module)
Minkowski property	?

$\rightsquigarrow$ the essential exponents of $V_w(\lambda)$ do not even make sense as before

Demazure modules: twisting

Fix

Twist everything by w^{-1} , so that $v_{\lambda \cdot w}$ becomes a highest weight vector of weight λ .

Demazure modules: twisting

Fix

Twist everything by w^{-1} , so that $v_{\lambda \cdot w}$ becomes a highest weight vector of weight λ .

- weights: $\mu \mapsto \mu \cdot w^{-1}$, so the generator gets weight $(\lambda \cdot w) \cdot w^{-1} = \lambda$;
the twisted Demazure character provides all weight space dimensions

Demazure modules: twisting

Fix

Twist everything by w^{-1} , so that $v_{\lambda \cdot w}$ becomes a highest weight vector of weight λ .

- weights: $\mu \mapsto \mu \cdot w^{-1}$, so the generator gets weight $(\lambda \cdot w) \cdot w^{-1} = \lambda$;
the twisted Demazure character provides all weight space dimensions
- operators: replace the birational sequence $(\beta_1, \dots, \beta_N)$ by $(\beta_1 \cdot w^{-1}, \dots, \beta_N \cdot w^{-1})$

Demazure modules: twisting

Fix

Twist everything by w^{-1} , so that $v_{\lambda \cdot w}$ becomes a highest weight vector of weight λ .

- weights: $\mu \mapsto \mu \cdot w^{-1}$, so the generator gets weight $(\lambda \cdot w) \cdot w^{-1} = \lambda$;
the twisted Demazure character provides all weight space dimensions
- operators: replace the birational sequence $(\beta_1, \dots, \beta_N)$ by $(\beta_1 \cdot w^{-1}, \dots, \beta_N \cdot w^{-1})$
- the Minkowski property survives for fixed w : $\text{es}_w(\lambda) + \text{es}_w(\mu) \subseteq \text{es}_w(\lambda + \mu)$

Demazure modules: twisting

Fix

Twist everything by w^{-1} , so that $v_{\lambda \cdot w}$ becomes a highest weight vector of weight λ .

- weights: $\mu \mapsto \mu \cdot w^{-1}$, so the generator gets weight $(\lambda \cdot w) \cdot w^{-1} = \lambda$;
the twisted Demazure character provides all weight space dimensions
- operators: replace the birational sequence $(\beta_1, \dots, \beta_N)$ by $(\beta_1 \cdot w^{-1}, \dots, \beta_N \cdot w^{-1})$
- the Minkowski property survives for fixed w : $\text{es}_w(\lambda) + \text{es}_w(\mu) \subseteq \text{es}_w(\lambda + \mu)$

In the twisted picture the algorithm applies **verbatim**:

Minkowski gluing, weight-space patching, tensor-product model — untwist for the output.

Demazure modules: twisting

Fix

Twist everything by w^{-1} , so that $v_{\lambda \cdot w}$ becomes a highest weight vector of weight λ .

- weights: $\mu \mapsto \mu \cdot w^{-1}$, so the generator gets weight $(\lambda \cdot w) \cdot w^{-1} = \lambda$;
the twisted Demazure character provides all weight space dimensions
- operators: replace the birational sequence $(\beta_1, \dots, \beta_N)$ by $(\beta_1 \cdot w^{-1}, \dots, \beta_N \cdot w^{-1})$
- the Minkowski property survives for fixed w : $\text{es}_w(\lambda) + \text{es}_w(\mu) \subseteq \text{es}_w(\lambda + \mu)$

In the twisted picture the algorithm applies **verbatim**:

Minkowski gluing, weight-space patching, tensor-product model — untwist for the output.

Sanity check: $w = w_0$ recovers the highest-weight algorithm.

Demazure modules in OSCAR

```
julia> basis_lie_demazure(:A, 2, [1, 0], [1])
```

Monomial basis of a Demazure module

of extremal weight $(w_1) * s_1$

of dimension 2

with monomial ordering degrevlex([x1, x2, x3])

over abstract Lie algebra of type A2 over QQ

where the used birational sequence consists of the following roots:

[a_1, a_2, a_1 + a_2]

- **one code path** for highest-weight and Demazure modules — only the twist and the character formula differ

Demazure modules in OSCAR

```
julia> basis_lie_demazure(:A, 2, [1, 0], [1])
Monomial basis of a Demazure module
  of extremal weight (w_1) * s1
  of dimension 2
  with monomial ordering degrevlex([x1, x2, x3])
over abstract Lie algebra of type A2 over QQ
where the used birational sequence consists of the following roots:
[a_1, a_2, a_1 + a_2]
```

- **one code path** for highest-weight and Demazure modules — only the twist and the character formula differ
- same variants and customization options: `basis_lie_demazure_string`, `..._lusztig`, `..._nz`, `..._ffl`

Toric degenerations of flag varieties

- $G/B \hookrightarrow \mathbb{P}(V(\lambda))$ for regular dominant λ , with homogeneous coordinate ring $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$

Toric degenerations of flag varieties

- $G/B \hookrightarrow \mathbb{P}(V(\lambda))$ for regular dominant λ , with homogeneous coordinate ring $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$
- monomial bases of all $V(k\lambda) \rightsquigarrow$ degenerate $\mathbb{C}[G/B]$ to its **initial algebra**: a flat family over $\mathbb{C}$ with generic fibre G/B and **toric** special fibre (encoded by a Newton–Okounkov body $\Delta(S, \prec, \lambda)$)

Toric degenerations of flag varieties

- $G/B \hookrightarrow \mathbb{P}(V(\lambda))$ for regular dominant λ , with homogeneous coordinate ring $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$
- monomial bases of all $V(k\lambda) \rightsquigarrow$ degenerate $\mathbb{C}[G/B]$ to its **initial algebra**: a flat family over $\mathbb{C}$ with generic fibre G/B and **toric** special fibre (encoded by a Newton–Okounkov body $\Delta(S, \prec, \lambda)$)
— analogously for Demazure modules and Schubert varieties

Toric degenerations of flag varieties

- $G/B \hookrightarrow \mathbb{P}(V(\lambda))$ for regular dominant λ , with homogeneous coordinate ring $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$
- monomial bases of all $V(k\lambda) \rightsquigarrow$ degenerate $\mathbb{C}[G/B]$ to its **initial algebra**: a flat family over $\mathbb{C}$ with generic fibre G/B and **toric** special fibre (encoded by a Newton–Okounkov body $\Delta(S, \prec, \lambda)$)
— analogously for Demazure modules and Schubert varieties

Conjecture (Fang, Fourier, Littelmann, 2017)

For **every** birational sequence S and monomial order $\prec$, the special fibre is an affine toric variety (equivalently, $\Delta(S, \prec, \lambda)$ is a rational polytope).

Toric degenerations of flag varieties

- $G/B \hookrightarrow \mathbb{P}(V(\lambda))$ for regular dominant λ , with homogeneous coordinate ring $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$
- monomial bases of all $V(k\lambda) \rightsquigarrow$ degenerate $\mathbb{C}[G/B]$ to its **initial algebra**: a flat family over $\mathbb{C}$ with generic fibre G/B and **toric** special fibre (encoded by a Newton–Okounkov body $\Delta(S, \prec, \lambda)$)
— analogously for Demazure modules and Schubert varieties

Conjecture (Fang, Fourier, Littelmann, 2017)

For **every** birational sequence S and monomial order $\prec$, the special fibre is an affine toric variety (equivalently, $\Delta(S, \prec, \lambda)$ is a rational polytope).

Known for special choices: (Fang, Fourier, Littelmann, 2017; Fujita, Naito, 2017)
e.g. type A/C with a height-descending sequence and neglex $\rightsquigarrow \Delta$ is a FFLV polytope;
similarly: string, Nakashima–Zelevinsky, Lusztig polytopes

Toric degenerations of flag varieties

- $G/B \hookrightarrow \mathbb{P}(V(\lambda))$ for regular dominant λ , with homogeneous coordinate ring $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$
- monomial bases of all $V(k\lambda) \rightsquigarrow$ degenerate $\mathbb{C}[G/B]$ to its **initial algebra**: a flat family over $\mathbb{C}$ with generic fibre G/B and **toric** special fibre (encoded by a Newton–Okounkov body $\Delta(S, \prec, \lambda)$)
— analogously for Demazure modules and Schubert varieties

Conjecture (Fang, Fourier, Littelmann, 2017)

For **every** birational sequence S and monomial order $\prec$, the special fibre is an affine toric variety (equivalently, $\Delta(S, \prec, \lambda)$ is a rational polytope).

Known for special choices: (Fang, Fourier, Littelmann, 2017; Fujita, Naito, 2017)
e.g. type A/C with a height-descending sequence and neglex $\rightsquigarrow \Delta$ is a FFLV polytope;
similarly: string, Nakashima–Zelevinsky, Lusztig polytopes
 $\rightsquigarrow$ **compute** $\text{es}(\lambda)$ for many choices of $(S, \prec)$ to explore and test the conjecture

The conjecture, made computable

Recall the conjecture: is the special fibre of the degeneration of $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$ always an affine toric variety?

The conjecture, made computable

Recall the conjecture: is the special fibre of the degeneration of $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$ always an affine toric variety?

Conjecture, reformulated

Is there an $N > 0$ such that for all $k \geq 0$, the set $\text{es}((N + k)\lambda)$ is a Minkowski sum of the sets $\text{es}(n\lambda)$ with $n < N$?

The conjecture, made computable

Recall the conjecture: is the special fibre of the degeneration of $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$ always an affine toric variety?

Conjecture, reformulated

Is there an $N > 0$ such that for all $k \geq 0$, the set $\text{es}((N + k)\lambda)$ is a Minkowski sum of the sets $\text{es}(n\lambda)$ with $n < N$?

- best case $N = 1$: $\text{es}(k\lambda) = \text{es}(\lambda) + \cdots + \text{es}(\lambda)$ — everything is generated in degree 1

The conjecture, made computable

Recall the conjecture: is the special fibre of the degeneration of $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$ always an affine toric variety?

Conjecture, reformulated

Is there an $N > 0$ such that for all $k \geq 0$, the set $\text{es}((N + k)\lambda)$ is a Minkowski sum of the sets $\text{es}(n\lambda)$ with $n < N$?

- best case $N = 1$: $\text{es}(k\lambda) = \text{es}(\lambda) + \cdots + \text{es}(\lambda)$ — everything is generated in degree 1
- `basis_coordinate_ring_kodaira` computes $\text{es}(k\lambda)$ degree by degree and returns the monomials **not** obtained from lower degrees $\rightsquigarrow$ it searches for N

The conjecture, made computable

Recall the conjecture: is the special fibre of the degeneration of $\mathbb{C}[G/B] \cong \bigoplus_{k \geq 0} V(k\lambda)^*$ always an affine toric variety?

Conjecture, reformulated

Is there an $N > 0$ such that for all $k \geq 0$, the set $\text{es}((N + k)\lambda)$ is a Minkowski sum of the sets $\text{es}(n\lambda)$ with $n < N$?

- best case $N = 1$: $\text{es}(k\lambda) = \text{es}(\lambda) + \cdots + \text{es}(\lambda)$ — everything is generated in degree 1
- `basis_coordinate_ring_kodaira` computes $\text{es}(k\lambda)$ degree by degree and returns the monomials **not** obtained from lower degrees $\rightsquigarrow$ it searches for N
- the same works for Schubert varieties:
`basis_coordinate_ring_kodaira_demazure`

Same variety, different choices

Example: G_2 , $\lambda = \omega_1$ ($\dim V(\lambda) = 7$; here G/B becomes a partial flag variety G/P).

Same variety, different choices

Example: G_2 , $\lambda = \omega_1$ ($\dim V(\lambda) = 7$; here G/B becomes a partial flag variety G/P).
 Number of monomials in $\text{es}(k\lambda)$ that are not obtained from lower degrees:

degree k	1	2	3	4	5	6

Same variety, different choices

Example: G_2 , $\lambda = \omega_1$ ($\dim V(\lambda) = 7$; here G/B becomes a partial flag variety G/P).
Number of monomials in $\text{es}(k\lambda)$ that are not obtained from lower degrees:

degree k	1	2	3	4	5	6
all pos. roots by desc. height, degrevlex	7	0	0	0	0	0

- first choice: $N = 1$, generated in degree 1 $\rightsquigarrow$ flat toric degeneration of G/P

Same variety, different choices

Example: G_2 , $\lambda = \omega_1$ ($\dim V(\lambda) = 7$; here G/B becomes a partial flag variety G/P).
Number of monomials in $\text{es}(k\lambda)$ that are not obtained from lower degrees:

degree k	1	2	3	4	5	6
all pos. roots by desc. height, degrevlex	7	0	0	0	0	0
all pos. roots by inc. height, invlex	7	5	14	7	12	8

- first choice: $N = 1$, generated in degree 1 $\rightsquigarrow$ flat toric degeneration of G/P
- second choice: new monomials in every degree tested (up to degree 14) — no N in sight

Same variety, different choices

Example: G_2 , $\lambda = \omega_1$ ($\dim V(\lambda) = 7$; here G/B becomes a partial flag variety G/P).
Number of monomials in $\text{es}(k\lambda)$ that are not obtained from lower degrees:

degree k	1	2	3	4	5	6
all pos. roots by desc. height, degrevlex	7	0	0	0	0	0
all pos. roots by inc. height, invlex	7	5	14	7	12	8

- first choice: $N = 1$, generated in degree 1 $\rightsquigarrow$ flat toric degeneration of G/P
- second choice: new monomials in every degree tested (up to degree 14) — no N in sight

Whether the conjecture becomes visible depends strongly on the choice of $(S, \prec)$.

Summary

- essential exponents give **uniform, computable** monomial bases of highest-weight *and* Demazure modules

Summary

- essential exponents give **uniform, computable** monomial bases of highest-weight *and* Demazure modules
- divide and conquer: glue by **Minkowski sums**, patch weight spaces with **polyhedral geometry** where needed

Summary

- essential exponents give **uniform, computable** monomial bases of highest-weight *and* Demazure modules
- divide and conquer: glue by **Minkowski sums**, patch weight spaces with **polyhedral geometry** where needed
- implemented in OSCAR (`experimental/BasisLieHighestWeight`)

Summary

- essential exponents give **uniform, computable** monomial bases of highest-weight *and* Demazure modules
- divide and conquer: glue by **Minkowski sums**, patch weight spaces with **polyhedral geometry** where needed
- implemented in OSCAR (`experimental/BasisLieHighestWeight`)
- a tool to **explore** the toric-degeneration conjecture

Summary

- essential exponents give **uniform, computable** monomial bases of highest-weight *and* Demazure modules
- divide and conquer: glue by **Minkowski sums**, patch weight spaces with **polyhedral geometry** where needed
- implemented in OSCAR (`experimental/BasisLieHighestWeight`)
- a tool to **explore** the toric-degeneration conjecture
- ongoing: performance tuning, in particular the integer programming formulation

Summary

- essential exponents give **uniform, computable** monomial bases of highest-weight *and* Demazure modules
- divide and conquer: glue by **Minkowski sums**, patch weight spaces with **polyhedral geometry** where needed
- implemented in OSCAR (`experimental/BasisLieHighestWeight`)
- a tool to **explore** the toric-degeneration conjecture
- ongoing: performance tuning, in particular the integer programming formulation

Read more

X. Fang, G. Fourier, L. Götting, and B. Wilop. *Computing monomial bases in Lie theory using OSCAR*. 2024. arXiv: 2403.15018 [math.RT]

Demazure part: paper in preparation

Summary

- essential exponents give **uniform, computable** monomial bases of highest-weight *and* Demazure modules
- divide and conquer: glue by **Minkowski sums**, patch weight spaces with **polyhedral geometry** where needed
- implemented in OSCAR (`experimental/BasisLieHighestWeight`)
- a tool to **explore** the toric-degeneration conjecture
- ongoing: performance tuning, in particular the integer programming formulation

Read more

X. Fang, G. Fourier, L. Göttgens, and B. Wilop. *Computing monomial bases in Lie theory using OSCAR*. 2024. arXiv: 2403.15018 [math.RT]

Demazure part: paper in preparation

Give it a try: [🔗 oscar-system.org](https://oscar-system.org)

References

- [CFL23] R. Chirivì, X. Fang, and P. Littelmann. “Seshadri stratifications and standard monomial theory”. In: *Inventiones mathematicae* 234 (2023), pp. 489–572.
- [Dem74] M. Demazure. “Une nouvelle formule des caractères”. In: *Bull. Sci. Math. (2)* 98.3 (1974), pp. 163–172.
- [FFGW24] X. Fang, G. Fourier, L. Göttgens, and B. Wilop. *Computing monomial bases in Lie theory using OSCAR*. 2024. arXiv: 2403.15018 [math.RT].
- [FFL11] E. Feigin, G. Fourier, and P. Littelmann. “PBW filtration and bases for irreducible modules in type A_n ”. In: *Transform. Groups* 16.1 (2011), pp. 71–89.
- [FFL17] X. Fang, G. Fourier, and P. Littelmann. “Essential bases and toric degenerations arising from birational sequences”. In: *Adv. Math.* 312 (2017), pp. 107–149.
- [FN17] N. Fujita and S. Naito. “Newton-Okounkov convex bodies of Schubert varieties and polyhedral realizations of crystal bases”. In: *Math. Z.* 285.1-2 (2017), pp. 325–352.
- [GHKK18] M. Gross, P. Hacking, S. Keel, and M. Kontsevich. “Canonical bases for cluster algebras”. In: *J. Amer. Math. Soc.* 31.2 (2018), pp. 497–608.
- [Kas93] M. Kashiwara. “Global crystal bases of quantum groups”. In: *Duke Math. J.* 69.2 (1993), pp. 455–485.
- [Lit98] P. Littelmann. “Contracting modules and standard monomial theory for symmetrizable Kac-Moody algebras”. In: *J. Amer. Math. Soc.* 11.3 (1998), pp. 551–567.
- [LS86] V. Lakshmibai and C. S. Seshadri. “Geometry of G/P . V”. In: *J. Algebra* 100.2 (1986), pp. 462–557.
- [Lus90] G. Lusztig. “Canonical bases arising from quantized enveloping algebras”. In: *J. Amer. Math. Soc.* 3.2 (1990), pp. 447–498.
- [MV88] I. Mirković and K. Vilonen. “Characteristic varieties of character sheaves”. In: *Invent. Math.* 93.2 (1988), pp. 405–418.

The full interface

Three function families: `basis_lie_highest_weight*`, `basis_lie_demazure*`, `basis_coordinate_ring_kodaira*`

variant	birational sequence	monomial ordering
(none)	all positive roots, increasing height	free (default: degrevlex)
<code>_ffl</code>	all positive roots, decreasing height	degrevlex
<code>_lusztig</code>	inversions of w_0 along a reduced expression	wdegrevlex (by root height)
<code>_string</code>	simple roots along a reduced expression of w_0	neglex
<code>_nz</code>	simple roots along a reduced expression of w_0	degrevlex

The base functions also accept custom birational sequences, given by indices of positive roots or by the roots themselves.

Polytopes of essential exponents

The **polytope of essential exponents** $P(\lambda) := \text{conv}(\text{es}(\lambda)) \subseteq \mathbb{R}^N$ is a degree-one approximation of the Newton–Okounkov body: if $\text{es}(k\lambda) = \text{es}(\lambda) + \cdots + \text{es}(\lambda)$ for all k , then $\Delta(S, \prec, \lambda) = P(\lambda)$.

```
julia> basis = basis_lie_highest_weight(:A, 3, [1,1,1];  
      monomial_ordering = :deglex, compute_polytope = true)
```

Monomial basis of a highest weight module

of highest weight $w_1 + w_2 + w_3$

of dimension 64

with monomial ordering deglex([x1, x2, x3, x4, x5, x6])

over abstract Lie algebra of type A3 over QQ

where the used birational sequence consists of the following roots:

[a_1, a_2, a_3, a_1 + a_2, a_2 + a_3, a_1 + a_2 + a_3]

```
julia> polytope_of_essential_exponents(basis)
```

Polyhedron in ambient dimension 6

Computable probes: is $\text{es}(\lambda) = P(\lambda) \cap \mathbb{Z}^N$? Is $P(\lambda) + P(\mu) = P(\lambda + \mu)$? Does $P(\lambda)$ match a known polytope?