

# QUIVERS AND MODULI OF THEIR THIN SINCERE REPRESENTATIONS IN MACAULAY2

MARY BARKER AND PATRICIO GALLARDO

**ABSTRACT.** We introduce the Macaulay2 package `ThinSincereQuivers` for studying acyclic quivers, the moduli of their thin-sincere representations, and the reflexive flow polytopes associated to them. We provide some background in the topic and illustrate how the package recovers examples from the literature.

## 1. INTRODUCTION

Our work, the package *ThinSincere Quivers* for Macaulay2 [GS], provides computational tools at the crossroads of toric geometry, graph theory, and moduli spaces. The main objects are acyclic quivers (finite directed graphs) and representations that associate a one-dimensional vector space to each vertex, i.e., a dimension vector equal to all 1's. These representations are called *thin sincere* ones and their moduli spaces are projective toric varieties associated to the well-studied flow polytopes, see [Hil98], [AH99], [Hil03] and [DJ16]. Additional motivation is given by Craw and Smith [CS08] who showed that every toric variety is the fine moduli space for stable thin representations of an appropriate quiver with relations.

Our package has many potential applications. It can be used to construct families of reflexive polytopes which play a central role in Mirror symmetry [ANSW09]. It explicitly describes the changes among the possible flow polytopes constructed from a given quiver [Hil03], and constructs the irreducible components of the core of toric hyperkähler varieties [HS02]. It also provides toric compactifications of moduli spaces such as  $M_{0,n}$  [BH20]. We will illustrate such applications while recovering examples from above references.

The main functionality is encapsulated in the `ToricQuiver` datatype. Quivers are equipped with either an integer number for each of the arrows, i.e., an integer flow, or an integer weight for the vertices. The type `ToricQuiver` can be constructed by either an incidence matrix and a vector of weights or a function that produces an acyclic quiver from an arbitrary graph by utilizing the Package *Graphs* [BIJ<sup>+</sup>] (§§2.1). Given a quiver, the user can study its sub-quivers and their stability (§§2.2), the space of regular flows and the chamber structure within the space of weights (§§3.1), the flow polytope associated to a given weight (§§4.1), and the moduli spaces of thin-sincere representations (§§4.2). We also include a function for detecting if a quiver is tight (an important technical assumption) and implement an algorithm for tightening it.

Next, we describe the structure of our package. We remark that the package documentation contains more examples which illustrate how the package can be used.

**1.1. Acknowledgement.** We would like to thank Daniel McKenzie and Eloísa Grifo for helpful conversations related to this work. P. Gallardo thanks Washington University at St Louis and the University of California, Riverside for their support and welcoming environment. Mary Barker would like to thank the department of mathematics at Washington University in St. Louis for their support and access to computational resources.

## 2. DESCRIPTION

Let  $Q$  be an acyclic finite directed graph i.e. an acyclic quiver. Denote the vertices by  $Q_0$  and the arrows by  $Q_1$ , see Figure 1. We will have an integer associated to each arrow. This function  $\mathbf{w} : Q_1 \rightarrow \mathbb{Z}$  is known as an *integral flow* of  $Q$ , and it is called *regular* if it is non-negative i.e.,  $\mathbf{w}(a) \geq 0$  for all  $a \in Q_1$ . The vertices also have constants associated to them. A function

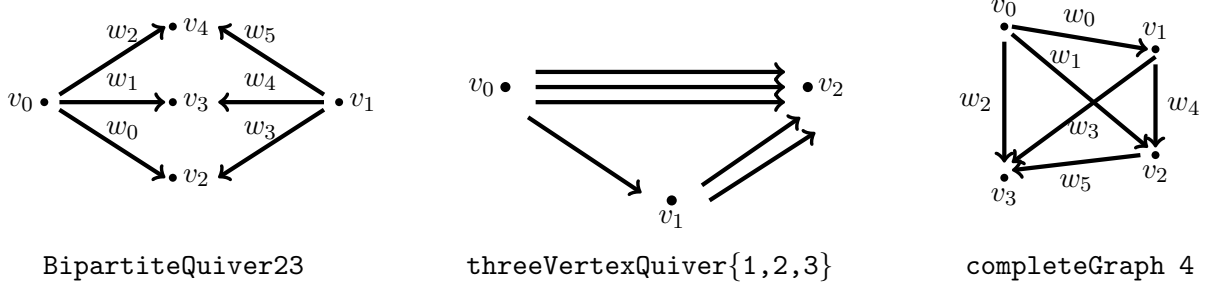


FIGURE 1. Examples of acyclic quivers

$\theta : Q_0 \rightarrow \mathbb{Z}$  is called an *integral weight* if  $\sum_{i \in Q_0} \theta(i) = 0$ . The set of all integral weights is denoted

as  $\text{Wt}(Q) \subset \mathbb{R}^{Q_0}$ . Every integral flow induces a weight on the vertices  $Q_0$  as follows: Let  $a \in Q_1$  be an arrow. Then,  $a^+ \in Q_0$  will denote its head and  $a^- \in Q_0$  will denote its tail. The so called *incidence map* generates a weight from the flow  $\mathbf{w}$  by

$$\text{inc}(\mathbf{w})(i) := \sum_{\substack{a \in Q_1 \\ a^+ = i}} \mathbf{w}(a) - \sum_{\substack{a \in Q_1 \\ a^- = i}} \mathbf{w}(a) \quad \text{for all } i \in Q_0.$$

In case the flow  $\mathbf{w}$  is not explicitly given, we assume that it is equal to  $\mathbf{1} := (1, \dots, 1)$ . The associated integral weight  $\delta_Q := \text{inc}(\mathbf{1})$  is called the *canonical weight*. If  $Q$  is connected, we have the following short exact sequence:

$$0 \rightarrow \text{Cir}(Q) \rightarrow \mathbb{Z}^{Q_1} \xrightarrow{\text{inc}} \text{Wt}(Q) \rightarrow 0$$

where the elements in  $\text{Cir}(Q)$  are called *integral circulations*. We remark that both the set of all integral circulations  $\text{Cir}(Q) \subset \mathbb{Z}^{Q_1}$  and the set of all integral weights  $\text{Wt}(Q) \subset \mathbb{Z}^{Q_0}$  forms a lattice within each space.

**2.1. Constructing quivers.** To define a quiver, we can enter the edges as pairs  $\{a, b\}$  where  $a$  and  $b$  are vertices and the arrow will take the orientation from  $a$  to  $b$ . The user can determine a particular flow, a random flow, or use the default flow of  $\mathbf{1} = (1, \dots, 1)$  by leaving it undeclared.

The main datatype in this package is a type of HashTable called *ToricQuiver*. It contains the following information: a list  $Q_0$  of integers representing the vertices, an ordered list  $Q_1$  of pairs of integers, representing the edges of the quiver, and the incidence matrix representation of the underlying directed graph. This is the  $|Q_0| \times |Q_1|$  matrix consisting of entries:

$$a_{i,j} = \begin{cases} 1, & \text{if vertex } i \text{ is the head of the } j^{\text{th}} \text{ edge} \\ -1, & \text{if vertex } i \text{ is the tail of the } j^{\text{th}} \text{ edge} \\ 0, & \text{otherwise} \end{cases}$$

In addition to these, the *ToricQuiver* datatype contains a list of integers representing the flow  $\mathbf{w}$  and the weight  $\text{inc}(\mathbf{w})$ .

**Example 2.1.** We generate the quiver  $Q$  associated to a bipartite graph, see left on Figure 1, with a random flow  $w$  as follows:

```
i1: loadPackage ("ThinSincereQuivers")
i2 : Q0 = {{0,2},{0,3},{0,4},{1,2},{1,3},{1,4}};
i3 : BipartiteQuiver23 = toricQuiver(Q0, Flow=>"Random")
o3 = ToricQuiver{connectivityMatrix => | -1 -1 -1 0 0 0 |
                                         | 0 0 0 -1 -1 -1 |
                                         | 1 0 0 1 0 0 |
                                         | 0 1 0 0 1 0 |
                                         | 0 0 1 0 0 1 |
                                         }
                                     flow => {71, 99, 69, 18, 46, 19}
                                     Q0 => {0, 1, 2, 3, 4}
                                     Q1 => {{0, 2}, {0, 3}, {0, 4}, {1, 2}, {1, 3}, {1, 4}}
```

We change the flow of the quiver BipartiteQuiver23 from its random value to another one  $w$  by

```
i1 : w = {1,1,1,1,1,1};
i2: BPw = toricQuiver(BipartiteQuiver23, w)
```

Given a quiver  $Q$  with weight  $\theta \in \text{Wt}(Q)$ , the user can recover its weight  $\theta$ , find a flow in the preimage  $\text{inc}^{-1}(\theta)$ , and detect if the underlying directed graph is acyclic by using the commands `theta(Q)`, `incInverse(th, Q)` where `th` =  $\theta$ , and `isAcyclic(Q)` respectively.

Additionally, given any connected graph  $G = (V, E)$  generated by the Package *Graphs* [BIJ<sup>+</sup>], we can construct an acyclic quiver  $Q$  by using the labeling of the vertices  $V$ . Indeed, the vertices  $V$  are indexed by integers  $i \in \{0, \dots, |V| - 1\}$ . Then, given an edge joining the vertices  $v_i$  and  $v_j$ , we set  $a^- := v_i$  if  $i < j$ . For example, to construct a quiver associated to the complete graph with four vertices, see [Hil03, Example 2], we use

```
i1: needsPackage ("Graphs")
i2: CG = toricQuiver completeGraph 4
```

Our package also has some built-in families. The command `bipartiteQuiver(r,n)` generates the quiver associated to the bipartite graph (for the case  $r = 2$  and  $n = 3$  see left in Figure 2). The type of quiver shown at the center of Figure 2 is generated by `threeVertexQuiver{a,b,c}` where  $a, b, c$  are positive integers (see Figure 1). For a list  $L = \{L_1, \dots, L_n\}$  of  $n$  positive integers, we define `chainQuiver {L_1, \dots, L_n}` as a linear chain with  $(n + 1)$  vertices. The entry  $L_i$  denotes the number of multiarrows between the vertices  $i$  and  $i + 1$ . Finally, one can combine two quivers to obtain a new one by identifying either a vertex or an arrow from each. These two operations are given in the functions `mergeOnVertex(Q1, V1, Q2, V2)` and `mergeOnArrow(Q1, A1, Q2, A2)` respectively.

**2.2. Subquivers and their stability.** A *subquiver*  $P$  of  $Q$  is defined as a quiver  $P$  such that  $P_0 \subset Q_0$ ,  $P_1 \subset Q_1$  and the flow function  $P_1 \rightarrow \mathbb{Z}$  is obtained by restricting the one from  $Q$ . The user has two options for working with subquivers. First, they can interpret the subquiver  $P \subset Q$  as one with the same vertices  $P_0 = Q_0$  but with a non-zero flow in the arrows  $P_1 \subsetneq Q_1$ . Since the arrows in  $Q_1$  are labeled, there is a subset  $I \subset \{0, \dots, |Q_1| - 1\}$  associated to  $P_1$ . The subquiver  $P$  can be constructed with the command `Q^I` (see Figure 2). The command `subquivers(Q)` lists all of the subquivers of  $Q$  with the format  $Q^I$ . The second option is to describe  $P$  without any reference to  $Q$ . This is done by the command `Q-I`. Finally, the list of arrows defining all spanning trees can be obtained with the command `allSpanningTrees(Q)`.

Given a weight  $\theta \in \text{Wt}(Q)$  and a subquiver  $P \subset Q$ , we can associate the concepts of  $\theta$ -stable,  $\theta$ -semistable, and  $\theta$ -unstable to  $P$ . They depend on a certain subset of vertices defined as follows:  $V \subset Q_0$  is called  $P$ -successor closed if there is no arrow in  $P_1$  leaving  $V$ . That is, for all  $a \in P_1$  with  $a^- \in V$ , we also have  $a^+ \in V$ . We can check if subset  $V \subset Q_0$  is  $P$ -successor closed by using

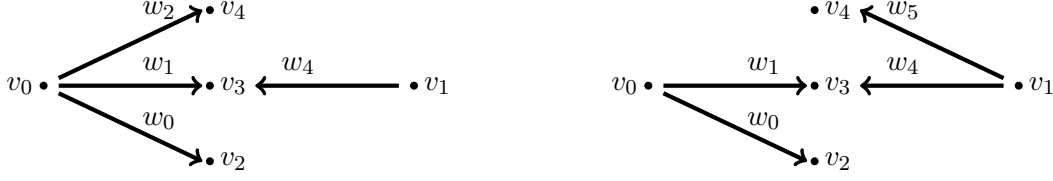


FIGURE 2. Subquivers of the bipartite quiver as defined in Example 2.1. They are labeled by the subsets  $I = \{0, 1, 2, 4\}$  and  $\{0, 1, 4, 5\}$

```

i1: Q = bipartiteQuiver(2,3)
i2: VA = {0,3}; VB = {1,3};
i3: P = Q~{0,1,2,4}
i4: isClosedUnderArrows(VA, P)
i5: isClosedUnderArrows(VB, P)
o4: false
o5: true

```

A subquiver  $P \subset Q$  is  $\theta$ -stable (resp.  $\theta$ -semistable) if and only if  $\sum_{i \in V} \theta(i) > 0$  (resp.  $\geq 0$ ) for all successor closed  $V$ .

Our package allows the user to test if a given subquiver  $P$  of  $Q$  is either  $\theta$ -stable or  $\theta$ -semistable by using

```

i1: Q = toricQuiver(bipartiteQuiver(2,3));
i2: P = Q~{0,1,4,5};
i3: isStable(P,Q)
i4: isSemistable(P,Q)

```

Notice that each quiver has always associated a weight that be can recovered with `theta(Q)`. Above commands test stability and semi-stability with respect to it.

A quiver is  $\theta$ -unstable if it is not  $\theta$ -semi-stable. If  $P \subset Q$  is a  $\theta$ -unstable quiver, then any quiver  $R \subset P$  is also  $\theta$ -unstable. Therefore, we can define *maximal  $\theta$ -unstable subquivers* as those that are not contained properly in any other  $\theta$ -unstable quiver. Given a quiver  $Q$  with weight  $\theta$ , we can assume any maximal unstable quiver  $P$  satisfies  $P_0 = Q_0$  and  $P_1 \subsetneq Q_1$ . Since  $P$  is determined by a subset  $I \subset Q_1$ . We enumerate all maximal  $\theta$ -unstable quivers by listing such subsets.

```

i1 : maximalUnstableSubquivers( bipartiteQuiver(2,3) )
o1: HashTable{NonSingletons => {{0, 1, 2, 3}, {0, 1, 2, 4}, {0, 1, 2, 5},
    {0, 3, 4, 5}, {1, 3, 4, 5}, {2, 3, 4, 5}}}

```

Similarly, if  $P \subset Q$  is not  $\theta$ -stable, then any quiver  $R \subset P$  is also not  $\theta$ -stable. Therefore, we can consider *maximal not  $\theta$  stable  $\theta$ -subquivers*. They are quivers that are not  $\theta$ -stable and are maximal with respect to the containment order. The command in this case is

```

i1: maximalNonstableSubquivers( bipartiteQuiver(2,3) ).

```

The output of above commands is in the form of a hash table. We remark that the list of unstable and not-stable quivers depends on the weight  $\theta$ . We illustrate such a difference for the quiver `bipartiteQuiver(2,3)` but an alternative weight equal to  $\{-5, -1, 2, 2, 2\}$ .

```

i1: w = incInverse({-5,-1,2,2,2},bipartiteQuiver(2,3)); -- find the flow
i2: Q = toricQuiver(bipartiteQuiver(2,3), w); -- change the flow
i3: maximalUnstableSubquivers(Q)
o3: HashTable{NonSingletons =>{{0, 1, 3, 4, 5},{0, 2, 3, 4, 5},{1, 2, 3, 4, 5}}}

```

We now turn to an important technical condition. A weight  $\theta$  is called *tight* if for every arrow  $\alpha \in Q_1$  the subquiver  $P$  with  $P_0 = Q_0$  and  $P_1 = Q_1 \setminus \{\alpha\}$  is  $\theta$ -stable. We can test this property as follows.

```

i1: CG = toricQuiver(completeGraph(4), {1, -2, 3, 0, 0, 0} );
i2: isTight(CG)
o1: false

```

The function "makeTight", returns a tight quiver such that the associated flow polytope (to be defined next section) does not change. The tightening process is outlined in [AVS09], see also [DJ16].

```

i1: CGb = makeTight({-2,1,-2,3},toricQuiver completeGraph 4)
o1: ToricQuiver{connectivityMatrix => | -1 -1 -1 -1 | }
                                   | 1 1 1 1 |
                                   flow => {-2, 1, 1, 1}
                                   Q0 => {0, 1}
                                   Q1 => {{0, 1}, {0, 1}, {0, 1}, {0, 1}}
                                   weights => {-1, 1}

i2: isTight(CGb)
o1: true

```

### 3. REGULAR FLOWS AND CONE OF WEIGHTS

We next turn to other combinatorial structures associated to our quiver. The main reference is [Hil03]. For a given weight  $\theta \in \text{Wt}(Q)$ , there is an integral polytope

$$\Delta(\theta) = \{\mathbf{w} \in \mathbb{R}^{Q_1} \mid \text{inc}(\mathbf{w}) = \theta\} \cap \mathbb{R}_{\geq 0}^{Q_1}$$

which is called a *flow polytope*. In general, given a weight  $\theta \in \text{Wt}(Q)$  the polytope  $\Delta(\theta)$  can be empty. However, there exists a *cone of weights*  $C(Q) \subset \mathbb{R}^{Q_0}$  such that  $\Delta(\theta)$  is not empty if and only if  $\theta \in C(Q)$ . The polytope has the expected dimension,  $|Q_1| - |Q_0| + 1$ , if and only if  $\theta$  is in the interior of  $C(Q)$ . The vertices of  $C(Q)$  are constructed with the so called primitive arrows from the quiver, see [Hil03, Prop. 4.7]. The user can recover the list of primitive arrows, the cone  $C(Q)$ , and test the membership of  $\theta$  in  $C(Q)$  with the following code

```

i1: needsPackage "Polyhedra"
i2: CG = toricQuiver completeGraph 4 -- the quiver
i3: A = primitiveArrows CG
i4: ConeCG = coneFromVData quiverConnectivityMatrix(CG^A) -- C(Q)
i5: CanonicalWeight = transpose matrix {{-3,-1,2,2}}; -- weight
i5: inInterior(CanonicalWeight,ConeCG)
o1: true

```

We can interpret  $\Delta(\theta)$  as parametrizing all possible regular flows with input equal to  $\theta$ . Each point at the interior of  $\Delta(\theta)$  parametrizes a flow with entries that are strictly positive. The vertices of  $\Delta(\theta)$  parametrize regular flows with input  $\theta$  whose support is a spanning tree, see Figure 3. The user can recover all such spanning trees  $T_i \subset Q$  by

```

i1: CG = toricQuiver completeGraph 4;
i2: tht= {-2,1,-1,2} -- weight
i3: stableTrees(tht, CG)
o1: {{4,0,5}, {0,5,3}, {0,5,2},{0,5,1}}

```

Each element of the output is a tuple  $\{b_1, \dots, b_m\}$  where  $b_k$  denotes the arrows of  $Q$  that define the  $\theta$ -stable tree. In particular, we recover examples such as [Hil03, Fig 10]. Finally, the regular flows supported on above trees can be found by using

```

i4: for i in stableTrees(tht, CG) do print incInverse(tht,CG^i)
o4: {1, 0, 1, 0, 0, 1}
    {2, 0, 0, 0, 1, 1}

```

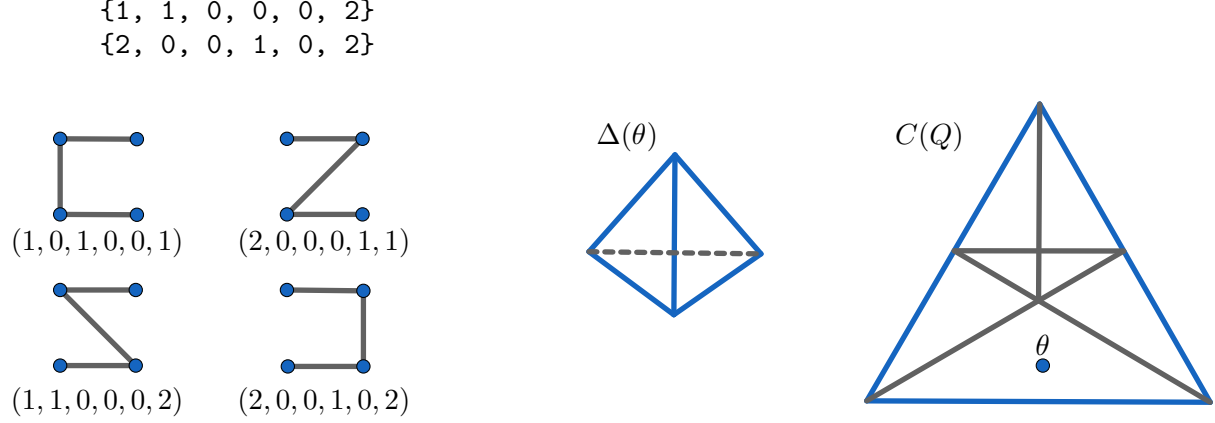


FIGURE 3. Cone of weights  $C(Q)$  (right), polytope  $\Delta(\theta)$  associated to the weight  $\theta = (-2, 1, -1, 2)$  (center) and  $\theta$ -stable trees parametrized by the vertices of the polytope  $\Delta(\theta)$  (left). Here, the quiver  $Q$  is constructed from the complete graph with four vertices.

**3.1. Walls and chambers on the space of weights.** The cone  $C(Q)$  has a wall-chamber decomposition induced by the following equivalence relation:  $\theta \sim \theta'$  if for every subquiver  $P \subset Q$ ,  $P$  is  $\theta$ -semistable implies  $P$  is  $\theta'$ -semistable and viceversa. In particular,  $\theta \sim t\theta$  for every  $t > 0$  and  $\Delta(\theta)$  is isomorphic to  $\Delta(\theta')$  up to affine lattice isomorphism. Our package allows the user to study this structure within  $C(Q)$  and recover it for given cases. We remark this problem with a different perspective and implementation has been solved in [BSDLV04, Sec 4]. For us, the starting point to describe the walls of this decomposition is the following result due to Hille.

**Lemma 3.1.** [Hil03, Sec 2.2] *Each wall  $W$  in  $C(Q)$  is contained in a hyperplane of the form*

$$W(Q_0^+) = \left\{ \theta \mid \sum_{i \in Q_0^+} \theta(i) = 0 \right\}$$

where  $Q = Q_0^+ \sqcup Q_0^-$  and the full subquivers  $Q^+$  and  $Q^-$  with vertices  $Q_0^+$  and  $Q_0^-$  are connected.

We denote any wall  $W(Q_0^+)$  by the subset of vertices  $Q_0^+$  used for defining it. The *type* of the wall  $W(Q_0^+)$  which is defined as  $(t^+, t^-)$  where  $t^+$  is the number of arrows starting  $Q_0^+$  and ending in  $Q_0^-$ , and  $t^-$  is the number of arrows starting  $Q^-$  and ending in  $Q^+$ . One can therefore describe a wall uniquely by the pair:  $Q_0^+, (t^+, t^-)$ .

```

i1: L = potentialWalls toricQuiver completeGraph 4;
i2: L#0
o1: Wall{Qplus => {1, 2, 3}}
    WallType => (0, 3)

```

A wall with  $t^+$  or  $t^-$  equal to 0 is called an outer wall. Otherwise, it is called an inner wall. By [Hil03, Lemma 4.5], facets of the cone  $C(Q)$  are of the form  $C(Q) \cap W$  where  $W$  is an outer wall. Although every wall is contained in a hyperplane  $W(Q_0)^+$ , there are contiguous chambers in which the hyperplane is not a “real wall” i.e., there exist two points  $x$  and  $y$  such that  $x \sim y$  and yet  $x$  and  $y$  are in different half-spaces defined by the hyperplane  $W$ . We can rely on the Package *Lattice Polytopes* [LS] for deciding whenever two polytopes are isomorphic (the required hypothesis that  $\Delta(\theta)$  is smooth is satisfied for  $\theta$  within the interior of a chamber).

```

i1: CG = toricQuiver completeGraph 4;
i1: sameChamber({-3,2,-1,2},{-2,1,-2,3}, CG)
o1: true

```

We remark that if the weights are not in  $C(Q)$  the flow polytopes will be empty.

```

i1: sameChamber({2,-1,1,-2},{3,-1,-1,-1}, CG)
o1: cannot be determined. stableTrees are empty

```

Another perspective on describing the chamber decomposition is also given by [Hil03]. Indeed, the spanning trees of  $Q$  induce chambers within  $C(Q)$  which are defined as

$$C_T = \text{inc}(D_P), \quad D_T = \{\mathbf{w} \in \mathbb{R}^{Q_1} \mid \mathbf{w}(a) = 0 \text{ for all } a \notin T_1, \mathbf{w}(a) \geq 0 \text{ for all } a \in T_1, \}.$$

A chamber  $C_T$  maybe the whole  $C(Q)$ , so they are not minimal with respect to the equivalence relation defined at the beginning of this section. However, the intersections  $\cap C_{T_i}$  which are of maximal dimension define a chamber system which either refines or equals the chamber decomposition of  $C(Q)$ , see [Hil03, Lemma 4.4]. The relevant commands to produce the refined chamber decomposition are

```

i1: CG = coneSystem(CG)          -- create the cone system
i2: rts = referenceThetas CG     -- returns a theta from each maximal chamber
o2: {{-1, -1, -1, 3}, {-2, 1, -1, 2}, {-3, 1, 1, 1}, {-2, -2, 1, 3},
     {-3, -1, 2, 2}, {-1, -3, 2, 2}, {-2, -2, 3, 1}}

```

This results recovers the chamber system for the quiver associated to the complete graph of four vertices, see [Hil03, Fig 7]. A similar computation recovers the 18 chambers associated to the bipartite graph with (2,3) vertices, see [HS02, Sec 9].

## 4. APPLICATIONS

We illustrate a few of the possible uses of our packages.

**4.1. To polytopes.** The polytopes  $\Delta(\theta)$  are a priori contained in  $\mathbb{R}^{Q_1}$  and the user can recover them formatted as a list of vertices with

```

i1 : flowPolytope({-3,-3,2,2,2}, bipartiteQuiver(2,3), Format=>"Original")
o2 = {{2, 0, 1, 0, 2, 1}, {2, 1, 0, 0, 1, 2}, {0, 2, 1, 2, 0, 1},
     {1, 2, 0, 1, 0, 2}, {0, 1, 2, 2, 1, 0}, {1, 0, 2, 1, 2, 0}}

```

However, a better representation can be obtained as follows. The polytope  $\Delta(\theta)$  is contained in the fiber  $\text{inc}^{-1}(\theta)$ . Therefore, we translate them to  $\text{Cir}(Q) = \text{inc}^{-1}(0)$ . This linear subspace is more natural because  $\dim \Delta(\theta) = \dim(\text{Cir}(Q)_{\mathbb{R}})$  for  $\theta$  in the interior of  $C(Q)$ . Moreover, there is a basis of the lattice  $\text{Cir}(Q)$  for each spanning tree  $T$  of  $Q$ . For example, given the quiver `bipartiteQuiver(2,3)` the following command computes a basis of the two dimensional vector space  $\text{Cir}_{\mathbb{R}}(Q) \subset \mathbb{R}^6$  using the spanning tree  $\{0, 1, 4, 5\}$ .

```

i1 : basisForFlowPolytope({0, 1, 4, 5}, bipartiteQuiver(2,3))
o1 = | 0  1  |
     | 1 -1  |
     | -1 0  |
     | 0 -1  |
     | -1 1  |
     | 1  0  |

```

The package automatically selects one if the none is provided and the user writes

```

i1: basisForFlowPolytope(bipartiteQuiver(2,3))|.

```

We can recover the vertices of  $\Delta(\theta)$  with respect to the basis induced by any spanning tree. For example, if  $Q$  is the bipartite graph and  $\theta$  is the canonical weight. Then,  $\Delta(\theta)$  is a hexagon. We obtain its vertices with respect to the basis induced by the stable tree with edges  $T = \{0, 1, 4, 5\}$ .

```
i8 : flowPolytope({-3,-3,2,2,2}, bipartiteQuiver(2,3), Format=>{0,1,4,5})
o8 = {{0, 1}, {1, 1}, {0, -1}, {1, 0}, {-1, -1}, {-1, 0}}
```

This output interfaces with other Macaulay2 packages such as `Polyhedra` [BLKa] and `OldPolyhedra` [Bir]. Then, it allows the reader to translate results based on toric Quiver computations into problems relating to polyhedral structures. For example, the flow polytope associated to the complete graph with 4 elements and  $\theta = \{-2, 1, -1, 2\}$  is isomorphic to the simplex

```
i0: needsPackage("LatticePolytopes");
i1 Q = toricQuiver completeGraph 4
i2: w = {-2, 1, -1, 2};
i3: FromK4 = flowPolytope(w, Q);
o3: {{0, 0, 0}, {0, 0, -1}, {-1, 0, 0}, {-1, 1, 0}}
```

We can also explore which polytopes  $\Delta(\theta)$  are reflexive. For example, given the quiver from the complete graph with four vertices and its canonical weight  $\{-3, -1, 1, 3\}$ , then we test if the associated polytope is reflexive.

```
i1: needsPackage ("OldPolyhedra");
i2: K4 = toricQuiver completeGraph 4;
i3: PolyK4 = convexHull transpose matrix flowPolytope({-3,-1,1,3}, K4);
i4: isReflexive PolyK4
o4: true
```

This result is expected because in fact the polytope  $\Delta(\delta_Q)$  is always reflexive [AH99, Prop 2.7].

**4.2. Applications to moduli spaces.** A thin sincere representation of a quiver  $Q$  assigns a one dimensional vector space to each vertex in  $Q_0$  and a linear map to each arrow in  $Q_1$ . The space of all representations of  $Q$  is  $\text{Rep}(Q) \cong \mathbb{C}^{Q_1}$ . Two representations  $\omega = (w_1, \dots, w_{|Q_1|})$  and  $\omega' = (w'_1, \dots, w'_{|Q_1|})$  are isomorphic to each other if there is an element  $\vec{v}_L = (t_1, \dots, t_{|Q_0|}) \in (\mathbb{C}^*)^{Q_0}$  such that  $w'_a t_{a^-} = t_{a^+} w_a$  for every arrow  $a \in Q_1$ . We observe that given a representation  $\omega$ , we can define a subquiver  $P := \text{Supp}(\omega)$  such that  $P_0 = Q_0$  and  $P_1 = \{a \in Q_1 \mid \omega(a) \neq 0\}$ . This quiver is called the support of  $\omega$ .

We say that a representation  $T$  is  $\theta$ -stable (resp.  $\theta$ -semistable) if for each proper subrepresentation  $S$  we have that  $\sum_{q \mid S(q) \neq 0} \theta(q) > 0$  ( $\geq 0$  resp). This notion of stability is equivalent to the one given in Section 2.2. Indeed, a representation  $T$  is  $\theta$ -stable precisely when the subquiver  $\text{Supp}(T)$  is  $\theta$ -stable, see [Hil98, Lemma 1.4] and discussion before [AH99, Sec 2.2]. By using tools from Geometric Invariant Theory, given a quiver  $Q$  with weight  $\theta \in C(Q)$ , there is a projective complex toric variety

$$\overline{M}(Q, \theta) := \text{Rep}(Q) //_{\theta} (\mathbb{C}^*)^{|Q_0| - 1}$$

of dimension  $|Q_1| - |Q_0| + 1$  parametrizing  $\theta$ -semistable thin-sincere representations up to isomorphism, see [Kin94] and [Hil98].

**Theorem 4.1.** [Hil98] *If  $\theta$  is in the interior of  $C(Q)$ , the compact toric variety  $\overline{M}(Q, \theta)$  is not empty and its associated polytope is equal to  $\Delta(\theta)$ .*

Within moduli theory it is important to describe the representations parametrized by  $\overline{M}(Q, \theta)$  as well as its geometry. Above discussion implies that we can describe the  $\theta$ -stable and  $\theta$ -semistable representations parametrized by every  $\overline{M}(Q, \theta)$ . Moreover, via toric geometry, our package can be used to study these moduli spaces. In particular, if  $\theta \sim \theta'$  then  $\overline{M}(Q, \theta)$  and  $\overline{M}(Q, \theta')$  are isomorphic. Therefore, the geometry of  $C(Q)$  describes the birational geometry of these compactifications. Particular examples of algebraic varieties constructed via our package include the blow up of  $\mathbb{P}^n$  along a linear subspace, see [Qin18], and toric compactifications of the moduli space of  $n$  labeled points in  $\mathbb{P}^1$ , see [BH20].

# REFERENCES

- [AH99] Klaus Altmann and Lutz Hille. Strong exceptional sequences provided by quivers. *Algebras and Representation Theory*, 2(1):1–17, 1999.
- [ANSW09] Klaus Altmann, Benjamin Nill, Sabine Schwentner, and Izolda Wiercinska. Flow polytopes and the graph of reflexive polytopes. *Discrete mathematics*, 309(16):4992–4999, 2009.
- [AVS09] Klaus Altmann and Duco Van Straten. Smoothing of quiver varieties. *manuscripta mathematica*, 129(2):211–230, 2009.
- [BH20] Mark Blume and Lutz Hille. Quivers and moduli spaces of pointed curves of genus zero, 2020.
- [BIJ<sup>+</sup>] Jack Burkart, David Cook II, Caroline Jansen, Amelia Taylor, and Augustine O’Keefe. Graphs: graphs and directed graphs (digraphs). Version 0.3.2. A *Macaulay2* package available at <https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages>.
- [Bir] René Birkner. OldPolyhedra: Convex polyhedra. Version 1.3. A *Macaulay2* package available at <https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages>.
- [BLKa] René Birkner and Lars Kastner (Maintaining author). Polyhedra: convex polyhedra. Version 1.10. A *Macaulay2* package available at <https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages>.
- [BSDLV04] Welleda Baldoni-Silva, Jesús A De Loera, and Michele Vergne. Counting integer flows in networks. *Foundations of Computational Mathematics*, 4(3):277–314, 2004.
- [CS08] Alastair Craw and Gregory G Smith. Projective toric varieties as fine moduli spaces of quiver representations. *American journal of mathematics*, 130(6):1509–1534, 2008.
- [DJ16] Mátyás Domokos and Dániel Joó. On the equations and classification of toric quiver varieties. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, 146(2):265–295, 2016.
- [GS] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at <http://www.math.uiuc.edu/Macaulay2/>.
- [Hil98] Lutz Hille. Toric quiver varieties. *Algebras and modules, II (Geiranger, 1996)*, 24:311–325, 1998.
- [Hil03] Lutz Hille. Quivers, cones and polytopes. *Linear algebra and its applications*, 365:215–237, 2003.
- [HS02] Tamás Hausel and Bernd Sturmfels. Toric hyperkähler varieties. *Documenta Mathematica*, 7:495–534, 2002.
- [Kin94] Alastair D King. Moduli of representations of finite dimensional algebras. *The Quarterly Journal of Mathematics*, 45(4):515–530, 1994.
- [LS] Anders Lundman and Gustav Sædén Ståhl. LatticePolytopes: A *Macaulay2* package. Version 1.0. A *Macaulay2* package available at <https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages>.
- [Qin18] Xuqiang Qin. Blow ups of  $\mathbb{P}^n$  as quiver moduli for exceptional collections. *arXiv preprint arXiv:1804.09544*, 2018.

DEPARTMENT OF MATHEMATICS AND STATISTICS, WASHINGTON UNIVERSITY IN ST. LOUIS, ONE BROOKINGS DRIVE, ST. LOUIS, MO, 63130  
*Email address:* marybarker@wustl.edu

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, RIVERSIDE, CA, 92521  
*Email address:* pgallard@ucr.edu